# TRUST-STATE STANDARD™ V2.5.1

## Deterministic Admissibility, Execution Governance, and Verifiable Trust-State

Published by
VTI Foundation, Inc.

Version: 2.5.1
Document ID: TSS-2026-v2.5.1
Status: Published Standard
Classification: Normative Standard
Publication Date: August 21, 2026

Supersedes: Trust-State Standard™ v2.5.1

**Preface**

The Trust-State Standard™ defines a deterministic governance framework for consequence-bearing digital action.

The Standard establishes requirements for determining whether a proposed digital action is admissible, whether that action is eligible to proceed into execution, whether the governing conditions supporting that determination remain valid over time, and whether the resulting determination can be independently verified, recomputed, or replayed.

Trust-State governance is action-specific and condition-specific.

The Standard distinguishes identity, access, capability, authority, evidence, Rule-State, admissibility, Trust-State, and execution eligibility as separate governance conditions. It requires those conditions to remain identifiable, bounded, version-aware, and independently verifiable where required by the applicable Conformance Profile.

The Trust-State Standard is technology-neutral and jurisdiction-neutral. It does not prescribe substantive policy, determine whether a consequence is desirable, or establish legal or regulatory compliance. Instead, it specifies how declared governing conditions are deterministically evaluated and enforced at the boundary before consequence-bearing execution occurs.

Version 2.5.1 consolidates the Standard around three core Conformance Profiles:

**Profile I — Deterministic Admissibility**
Determines whether a proposed consequence-bearing action satisfies applicable governing conditions.

**Profile II — Deterministic Enforcement**
Determines admissibility and prevents non-eligible actions from crossing the execution boundary.

**Profile III — Deterministic Continuity and Recomputation**
Determines admissibility, enforces execution, governs continued validity across material change, and supports independent recomputation and replay.

The Standard is stewarded and published by VTI Foundation, Inc.

**Abstract**

The Trust-State Standard™ specifies deterministic requirements governing the conditions under which consequence-bearing digital actions become admissible, executable, and independently verifiable.

The Standard defines requirements for Authority-State, evidence sufficiency, Rule-State identity, contextual evaluation, admissibility determination, Trust-State derivation, execution eligibility, pre-execution enforcement, continuity, revocation, canonical serialization, integrity protection, independent recomputation, and replay.

The Standard establishes three Conformance Profiles corresponding to deterministic admissibility, deterministic enforcement, and deterministic continuity and recomputation.

A conforming implementation is not required to use a particular software architecture, programming language, policy engine, identity system, cryptographic implementation, artificial intelligence model, or execution environment. Conformance is determined by demonstrated satisfaction of the normative requirements applicable to the declared Conformance Profile and conformance boundary.

The Trust-State Standard does not establish whether governing policies are substantively correct, lawful, ethical, or desirable. It establishes whether the declared governing conditions applicable to a consequence-bearing digital action can be deterministically evaluated, enforced where required, preserved, and independently verified.

**Canonical Scope Statement**

The Trust-State Standard governs the deterministic conditions under which consequence-bearing digital action becomes admissible, executable, and independently verifiable.

**Table of Contents**



**Normative Standard**

**Annexes**

## Section 1 — Scope

### 1.0 Purpose

This section defines the scope, applicability, and boundaries of the Trust-State Standard.

The Trust-State Standard specifies requirements for the deterministic evaluation, enforcement, preservation, and independent verification of governance conditions applicable to consequence-bearing digital actions, execution pathways, and state transitions.

This section is normative.

### 1.1 Scope of the Standard

The Trust-State Standard applies to systems that determine whether a proposed digital action, execution pathway, or state transition is admissible before consequence-bearing execution is permitted to occur.

The Standard addresses:

- authority evaluation;
- evidence sufficiency;
- Rule-State identity;
- contextual constraints;
- admissibility determination;
- Trust-State derivation;
- execution eligibility;
- pre-execution enforcement;
- continuity governance;
- revocation handling;
- deterministic serialization;
- evidence preservation;
- independent verification;
- recomputation; and
- replay-equivalent evaluation.

**1.2 Applicable System Types**

The Standard MAY apply to:

- software systems;
- artificial intelligence systems;
- autonomous agents;
- workflow engines;
- authorization systems;
- policy enforcement systems;
- identity and credentialing systems;
- compliance systems;
- transaction systems;
- distributed systems;
- human-machine systems; and
- other digital systems capable of initiating, permitting, influencing, or executing consequence-bearing action.

Applicability depends upon the presence of a consequence-bearing action or state transition subject to governance conditions.

**1.3 Consequence-Bearing Actions**

For purposes of this Standard, a consequence-bearing action is any digital action, execution pathway, or state transition capable of materially affecting:

- a person;
- an organization;
- a resource;
- a legal or regulatory condition;
- a financial position;
- a credential;
- an entitlement;
- a permission;

- a system state;
- a security condition;
- a contractual state; or
- another governed interest.

The Standard does not require that consequence be harmful, irreversible, or high-risk.

The existence of governed consequence is sufficient to establish applicability.

**1.4 Pre-Execution Governance Boundary**

The principal governance boundary of the Trust-State Standard exists before consequence-bearing execution.

The Standard governs whether applicable authority, evidence, Rule-State, contextual conditions, and other admissibility requirements have been satisfied before execution eligibility is asserted.

Post-execution audit, monitoring, logging, or forensic reconstruction are outside the principal execution-governance boundary, except where such mechanisms are required to:

- preserve evidence;
- verify prior determinations;
- support recomputation;
- support continuity governance;
- establish accountability; or
- assess conformity.

**1.5 Technology Neutrality**

The Trust-State Standard is technology-neutral.

Conformance does not require a specific:

- software architecture;
- programming language;
- database;
- cryptographic implementation;
- identity protocol;

- distributed ledger;
- artificial intelligence model;
- cloud platform;
- hardware environment; or
- commercial product.

Implementations MAY differ provided that the applicable normative requirements are satisfied and independently verifiable.

**1.6 Jurisdictional Neutrality**

The Trust-State Standard is jurisdiction-neutral.

The Standard does not itself establish legal authority, regulatory approval, statutory compliance, or jurisdiction-specific authorization.

Jurisdictional requirements MAY be represented as Rule-State conditions, contextual constraints, authority conditions, evidence requirements, or other admissibility-governing inputs.

Claims of regulatory or statutory compliance SHALL be governed separately from claims of Trust-State conformance.

**1.7 Relationship to Risk, Governance, and Compliance**

The Trust-State Standard does not replace:

- enterprise risk management;
- cybersecurity governance;
- privacy governance;
- model risk management;
- legal compliance programs;
- regulatory supervision;
- organizational policy;
- professional judgment; or
- domain-specific safety controls.

The Standard provides a deterministic governance mechanism through which applicable conditions from such systems MAY be represented, evaluated, enforced, and independently verified at the execution boundary.

### 1.8 Non-Scope

The Trust-State Standard does not define:

- societal values;
- ethical outcomes;
- civilizational objectives;
- universally desirable consequences;
- legal rights;
- legal obligations;
- regulatory requirements;
- business policy;
- substantive authorization policy; or
- the correctness of external governance rules.

The Standard governs the deterministic application and verification of declared governing conditions.

It does not determine what those governing conditions ought to be.

### 1.9 Conformance Scope

Conformance SHALL be declared against a defined:

- implementation boundary;
- operational boundary;
- action class;
- Rule-State;
- evidence boundary;
- authority boundary;
- contextual boundary; and

- applicable Conformance Profile.

A conformance claim SHALL NOT imply conformity outside the declared boundary.

### 1.10 Canonical Scope Statement

The Trust-State Standard governs the deterministic conditions under which consequence-bearing digital action becomes admissible, executable, and independently verifiable.

### 1.11 Normative Language

The key words SHALL, SHOULD, and MAY in this Standard indicate the following:

SHALL indicates a mandatory requirement for conformity.

SHOULD indicates a recommended practice from which deviation is permitted when justified by the applicable implementation, Rule-State, Conformance Profile, or governance context.

MAY indicates a permitted option and does not establish a mandatory requirement.

The words MUST, REQUIRED, and equivalent normative terms SHOULD NOT be used to introduce requirements where SHALL is sufficient.

## Section 2 — Normative References

### 2.0 Purpose

This section identifies the normative references upon which requirements of the Trust-State Standard depend.

This section is normative.

### 2.1 Normative Reference Principle

A reference is normative only where conformance with the referenced material is necessary to satisfy a requirement of this Standard.

Informative publications, explanatory frameworks, regulatory mappings, examples, implementation guidance, and conceptual material SHALL NOT become normative solely by citation.

### 2.2 Trust-State Standard Annexes

Normative annexes formally designated as part of the applicable Trust-State Standard release SHALL constitute normative references where expressly incorporated by a requirement.

Each normative annex SHALL identify:

- its designation;
- version;
- normative status;
- applicable Standard version;
- scope; and
- relationship to conformance.

An implementation SHALL NOT be required to conform to an annex unless that annex is expressly applicable to the declared conformance boundary or Conformance Profile.

### 2.3 Deterministic Serialization Specification

Where deterministic serialization is required, the applicable Trust-State deterministic serialization specification SHALL govern:

- canonical representation;
- canonical ordering;

- canonical encoding;
- canonical hashing;
- representation boundaries; and
- reproducibility requirements.

**2.4 Conformance and Test Specifications**

Where formal conformance testing is required, the applicable Trust-State conformance specification, test suite, or conformance test specification SHALL govern the testable conditions for the declared Conformance Profile.

Test specifications SHALL NOT expand normative requirements beyond those established by the Standard and applicable normative annexes.

**2.5 External Standards**

External standards MAY be incorporated by reference where necessary to define an implementation requirement.

Where an external standard is incorporated normatively:

- the referenced edition or version SHALL be identified;
- the incorporated requirement SHALL be explicit;
- conflicts with the Trust-State Standard SHALL be resolved through the governance rules of the applicable release; and
- incorporation SHALL NOT imply adoption of unrelated provisions of the external standard.

**2.6 Regulatory and Legal Materials**

Laws, regulations, regulatory guidance, contractual requirements, and jurisdiction-specific requirements MAY inform Rule-State conditions or implementation requirements.

Such materials SHALL NOT become normative requirements of the Trust-State Standard unless expressly incorporated by the Standard.

Conformance with the Trust-State Standard SHALL NOT, by itself, constitute a claim of legal or regulatory compliance.

### 2.7 Consequence Science

Consequence Science provides the conceptual foundation for the Layer 0 principle and the relationship between consequence and governance.

Unless a specific Consequence Science requirement is expressly incorporated into a normative clause of this Standard, Consequence Science SHALL be treated as informative explanatory material rather than an independent source of conformance requirements.

### 2.8 Version Control

Normative references SHALL be version-identifiable.

A conforming implementation SHALL identify the versions of normative references relied upon in its declared conformance boundary.

Supersession of a normative reference SHALL NOT retroactively alter the requirements applicable to a previously declared conformance assessment unless expressly stated by the governing release.

## Section 3 — Terms and Definitions

### 3.0 Purpose

This section defines the canonical terminology used throughout the Trust-State Standard.

Unless otherwise stated, the terms in this section are normative.

### 3.1 Action

A proposed or initiated operation capable of changing, accessing, creating, transmitting, authorizing, modifying, or otherwise affecting a governed state, resource, interest, or condition.

### 3.2 Action Request

A request, instruction, command, transaction, message, invocation, or other input proposing that an action occur.

### 3.3 Admissibility

The condition in which a proposed consequence-bearing action, execution pathway, or state transition satisfies the governing requirements necessary for formation.

### 3.4 Admissibility Determination

The result of evaluating applicable admissibility-governing conditions for a proposed action.

### 3.5 Admissibility-Governing Conditions

The authority, evidence, Rule-State, policy, contextual, temporal, revocation, dependency, jurisdictional, and other objectively determinable conditions relevant to admissibility.

### 3.6 Authority

A verifiable source of permission, delegation, control, governance, jurisdiction, ownership, responsibility, or decision-making authority applicable to a proposed action.

### 3.7 Authority Lineage

The verifiable sequence through which authority originates, is delegated, constrained, modified, superseded, or revoked.

**3.8 Authority-State**

A version-identifiable representation of the authority conditions applicable to an actor, action, resource, scope, or execution context at a defined point or interval in time.

An Authority-State MAY include:

- authority source;
- delegation lineage;
- scope;
- constraints;
- provenance;
- temporal validity;
- assurance attributes;
- revocation state; and
- applicable identifiers.

**3.9 Canonical Representation**

A representation generated under defined deterministic rules such that equivalent inputs produce equivalent representations.

**3.10 Consequence**

An outcome resulting from an action, execution pathway, state transition, system behavior, or other consequence-bearing activity.

**3.11 Consequence-Bearing Action**

An action capable of materially affecting a governed person, organization, resource, state, entitlement, permission, obligation, security condition, financial condition, legal condition, or other governed interest.

**3.12 Consequence-Bearing State**

A state capable of initiating, permitting, influencing, enabling, or producing consequence-bearing execution.

**3.13 Context**

Objectively determinable conditions relevant to admissibility that are not fully represented by authority, evidence, or Rule-State alone.

**3.14 Deterministic Evaluation**

Evaluation in which equivalent governing inputs and equivalent Rule-State versions produce equivalent outputs under the declared evaluation procedure.

**3.15 Deterministic Representation**

A representation produced through canonicalization such that equivalent governed inputs produce an equivalent serialized result.

**3.16 Evidence**

Information, artifacts, records, credentials, attestations, observations, measurements, proofs, or other verifiable inputs used to support admissibility evaluation.

**3.17 Evidence Provenance**

Information sufficient to establish the origin, lineage, transformation, validation, and relevant custody history of evidence.

**3.18 Evidence Sufficiency**

The condition in which the evidence required by the applicable Rule-State is present, valid, and adequate for the proposed action.

**3.19 Execution Eligibility**

The condition in which a proposed consequence-bearing action or execution pathway is permitted to proceed under the applicable admissibility determination.

**3.20 Execution Enforcement**

A control that permits, prevents, terminates, or otherwise constrains execution based upon execution eligibility.

**3.21 Execution Boundary**

The point at which a proposed action transitions from evaluation into consequence-bearing execution.

**3.22 Independent Recomputation**

The reproduction of an admissibility, Trust-State, Authority-State, or execution eligibility determination by an independent evaluator using the relevant governing inputs, Rule-State, and deterministic evaluation procedure.

**3.23 Independent Verification**

Evaluation by a logically or operationally separate verifier capable of determining whether a claimed state, input, process, artifact, or outcome satisfies applicable requirements.

**3.24 Integrity Value**

A cryptographic or otherwise deterministic value derived from a canonical representation for the purpose of detecting modification or establishing representation identity.

**3.25 Material Change**

An objectively determinable change to a governing condition that may affect the validity of a prior admissibility, Trust-State, Authority-State, or execution eligibility determination.

**3.26 Pre-Execution Enforcement**

Execution enforcement applied before consequence-bearing execution is permitted to form.

**3.27 Replay**

Re-evaluation of a prior determination using preserved governing inputs and the applicable Rule-State or equivalent reconstruction thereof.

**3.28 Replay-Equivalent Evaluation**

An evaluation in which preserved equivalent governing conditions and equivalent Rule-State versions produce an equivalent deterministic result.

**3.29 Revocation**

The governed withdrawal, invalidation, or termination of validity from authority, evidence, credentials, delegation, Trust-State, execution eligibility, or another governed state or input.

**3.30 Rule**

A defined condition, constraint, requirement, prohibition, permission, transformation, or decision criterion applicable to evaluation.

**3.31 Rule Identity**

A stable identifier associated with a defined rule such that the rule can be independently distinguished from other rules and from modified versions of itself.

**3.32 Rule-State**

The version-identifiable set of rules, policies, constraints, governance requirements, and evaluation conditions applicable to an admissibility determination.

**3.33 Trust-State**

A deterministically derived condition representing whether applicable governance requirements for a declared action, scope, context, and Rule-State are satisfied.

Trust-State does not mean subjective trust, institutional reputation, confidence, reliability score, or generalized trustworthiness.

**3.34 Trust-State Continuity**

The condition in which the validity of a Trust-State is preserved or deterministically re-evaluated as governing conditions change over time.

**3.35 Trust-State Evidence**

The evidence sufficient to establish how a Trust-State determination was derived and, where required, to support independent verification or recomputation.

**3.36 Conformance**

Demonstrated satisfaction of all requirements applicable to a declared Trust-State Standard Conformance Profile and conformance boundary.

**3.37 Conformance Boundary**

The explicitly declared operational, implementation, action, authority, evidence, Rule-State, contextual, and temporal scope within which a conformance claim applies.

**3.38 Conformance Profile**

A formally defined set of Trust-State Standard requirements representing a particular level of deterministic admissibility, execution enforcement, continuity governance, verification, or recomputation capability.

**3.39 Certification**

A formal third-party determination that a defined implementation or system conforms to an identified Trust-State Standard release, Conformance Profile, and conformance boundary.

**3.40 Canonical Terminology Requirement**

Implementations claiming conformance SHALL use the defined meaning of terms in this section when those terms appear in conformance claims, certification evidence, normative mappings, implementation declarations, or audit materials.

Implementations MAY use additional domain-specific terminology provided that such terminology does not alter, obscure, or contradict the canonical meaning of terms defined by this Standard.

## Section 4 — Consequence and Admissibility Model

### 4.0 Purpose

This section establishes the foundational relationship between consequence, governance, admissibility, and execution under the Trust-State Standard.

The Trust-State Standard recognizes consequence as the foundational condition giving rise to governance requirements.

Governance does not create consequence.

Governance determines the conditions under which consequence-bearing actions, execution pathways, or state transitions may be permitted to occur.

This section is normative.

### 4.1 Consequence as the Foundational Condition

A consequence is an outcome resulting from an action, execution pathway, state transition, system behavior, or other consequence-bearing activity.

The existence or possibility of consequence creates the requirement for governance.

The Trust-State Standard therefore treats consequence as the foundational condition against which authority, evidence, applicable rules, contextual constraints, admissibility, and execution eligibility are evaluated.

Consequence SHALL NOT be interpreted as a system output that governance is required to guarantee.

Consequence SHALL instead define the boundary at which governance requirements become operationally relevant.

### 4.2 Layer 0 Principle

For purposes of the Trust-State Standard:

**Layer 0 is Consequence.**

Layer 0 represents the condition that a proposed action, execution pathway, state transition, or computational state is capable of producing consequence.

All Trust-State governance requirements arise above Layer 0 for the purpose of governing whether consequence-bearing activity is admissible.

Layer 0 SHALL NOT itself constitute an authorization mechanism, evidence source, rule-state, trust determination, or execution decision.

### 4.3 Governance Before Consequence

A conforming system SHALL evaluate applicable governance conditions before permitting formation of a consequence-bearing execution pathway where the declared conformance profile requires execution control.

Applicable governance conditions SHALL include, where relevant:

- authority;
- evidence;
- rule-state;
- policy constraints;
- contextual constraints;
- revocation status;
- dependency conditions;
- temporal conditions; and
- other objectively determinable admissibility conditions.

Post-execution logging, monitoring, audit, or reconstruction MAY support accountability and verification but SHALL NOT substitute for pre-execution admissibility evaluation where such evaluation is required by the applicable conformance profile.

### 4.4 Admissibility

Admissibility is the condition in which a proposed consequence-bearing action, execution pathway, or state transition satisfies the governing conditions necessary for formation.

Admissibility SHALL be derived from objectively determinable conditions.

An admissibility determination SHALL identify the governing inputs upon which the determination depends.

Where the applicable conformance profile requires deterministic admissibility, equivalent governing conditions and equivalent rule-state versions SHALL produce equivalent admissibility outcomes.

### 4.5 Admissibility-Governing Conditions

Admissibility-governing conditions SHALL include the conditions necessary to establish whether a proposed action may proceed.

These conditions MAY include:

- evidence sufficiency;
- authority sufficiency;
- rule sufficiency;
- rule-state identity;
- policy applicability;
- contextual constraints;
- temporal validity;
- revocation state;
- delegation lineage;
- dependency state;
- scope limitations;
- jurisdictional constraints; and
- other conditions declared by the governing rule-state.

A conforming implementation SHALL preserve sufficient information to identify which admissibility-governing conditions were evaluated.

**4.6 Authority**

Authority is a verifiable source of permission, delegation, control, governance, jurisdiction, ownership, responsibility, or decision-making capability applicable to a proposed action.

Authority SHALL be evaluated in relation to the specific action for which admissibility is being determined.

Possession of identity, access, credentials, information, or system capability SHALL NOT, by itself, establish sufficient authority to perform a consequential action.

Where authority is delegated, the relevant delegation lineage SHALL be verifiable.

Where authority is revoked, expired, superseded, constrained, or otherwise invalidated, a conforming system SHALL account for that state in admissibility evaluation.

### 4.7 Evidence

Evidence is information, artifacts, records, credentials, attestations, measurements, proofs, observations, or other verifiable inputs used to support admissibility evaluation.

A conforming system SHALL:

- identify evidence required by the applicable rule-state;
- determine whether required evidence is sufficient;
- preserve evidence provenance where required for verification;
- prevent assertion of admissibility where required evidence is absent or invalid; and
- support independent verification of evidence relied upon in the determination.

Evidence SHALL NOT be treated as sufficient solely because it exists.

Evidence sufficiency SHALL be determined in relation to the proposed action and governing rule-state.

### 4.8 Rule-State

Rule-State is the version-identifiable set of rules, policies, constraints, governance requirements, and evaluation conditions applicable to an admissibility determination.

A conforming system SHALL preserve sufficient information to establish which Rule-State governed a determination.

Rule-State identity SHALL be:

- version-identifiable;
- integrity-protected;
- attributable to a defined governance source; and
- resistant to unauthorized modification.

Equivalent admissibility determinations SHALL be evaluated against equivalent Rule-State versions.

### 4.9 Context

Context consists of objectively determinable conditions relevant to admissibility that are not fully represented by authority, evidence, or Rule-State alone.

Context MAY include:

- time;

- location or jurisdiction;
- operating environment;
- dependency state;
- system state;
- actor state;
- resource state;
- declared execution conditions; and
- other bounded conditions identified by the applicable Rule-State.

Context used in admissibility evaluation SHALL be representable in a manner sufficient to support verification and recomputation where required by the applicable conformance profile.

**4.10 Trust-State Determination**

A Trust-State is the deterministically derived condition representing whether the governing requirements applicable to a proposed action are satisfied.

A Trust-State SHALL be derived from the admissibility-governing conditions applicable to the proposed action.

Trust-State SHALL NOT be treated as institutional reputation, subjective confidence, generalized trustworthiness, or historical reputation.

A Trust-State determination SHALL be bounded to the declared action, scope, conditions, and Rule-State under which it was derived.

**4.11 Execution Eligibility**

Execution eligibility is the condition in which a proposed consequence-bearing action or execution pathway is permitted to proceed under the applicable Trust-State and Rule-State.

Where execution eligibility is required by the applicable conformance profile, a conforming system SHALL:

- derive execution eligibility from admissibility;
- prevent eligibility assertion absent required admissibility;
- preserve evidence of the eligibility determination; and
- support verification of the conditions upon which eligibility was based.

Execution eligibility SHALL NOT be inferred solely from possession of access rights, credentials, role membership, system capability, or prior authorization.

**4.12 Pre-Execution Enforcement**

Where required by the applicable conformance profile, a conforming system SHALL enforce execution eligibility before formation of a consequence-bearing execution pathway.

Such a system SHALL:

- prevent execution where admissibility has not been established;
- prevent execution where required authority is insufficient;
- prevent execution where required evidence is insufficient;
- prevent execution where applicable Rule-State conditions are unsatisfied;
- prevent reliance upon revoked or invalid governing inputs; and
- fail closed where required admissibility conditions cannot be established.

Pre-execution enforcement SHALL occur before consequence-bearing execution is permitted to form.

**4.13 Continuity of Admissibility**

Admissibility SHALL NOT be presumed to remain valid after a material change in governing conditions.

A material change MAY include change to:

- authority;
- evidence;
- Rule-State;
- revocation status;
- contextual constraints;
- dependency conditions;
- temporal validity; or
- other conditions material to the prior determination.

Where the applicable conformance profile requires continuity governance, the system SHALL detect material change and deterministically re-evaluate admissibility.

### 4.14 Revocation

A conforming system SHALL account for revocation where revocation is applicable to governing authority, evidence, credentials, Rule-State, delegation, or other admissibility-governing conditions.

The system SHALL prevent continued reliance upon a revoked governing input where that revocation affects admissibility.

Revocation state and revocation lineage SHALL be preserved where required to support verification, continuity, or recomputation.

### 4.15 Action-Level Governance

Admissibility SHALL be determined at a level sufficiently specific to the consequence-bearing action being evaluated.

System-level approval, model-level approval, application-level approval, identity-level approval, or generalized actor authorization SHALL NOT alone establish admissibility for a specific consequential action.

A conforming system SHALL support determination of whether the specific proposed action remains admissible under the governing conditions applicable at the time of evaluation.

### 4.16 Independent Verification and Recomputation

Where required by the applicable conformance profile, a conforming system SHALL preserve sufficient information to permit independent verification or recomputation of:

- governing evidence;
- governing authority;
- Rule-State identity;
- contextual conditions;
- admissibility determination;
- Trust-State determination; and
- execution eligibility determination.

Equivalent governing conditions and equivalent Rule-State versions SHALL produce equivalent deterministic outputs.

### 4.17 Probabilistic Inputs

Probabilistic methods MAY be used as supporting inputs to admissibility evaluation.

A probabilistic output SHALL NOT, by itself, establish admissibility where deterministic verification of governing conditions is required.

Where probabilistic inputs are used, the governing Rule-State SHALL define how those inputs are bounded, interpreted, and incorporated into the deterministic determination.

### 4.18 Conformance Principle

Conformance under this section is established through independently verifiable evidence demonstrating that the requirements applicable to the declared conformance profile have been satisfied.

Logging, monitoring, retrospective audit, confidence scoring, or probabilistic inference alone SHALL NOT establish conformity with requirements for deterministic pre-execution admissibility.

### 4.19 Canonical Statement

**Consequence is the reason governance exists.**

The Trust-State Standard governs the conditions under which consequence-bearing action becomes admissible.

Trust is established by determining what may proceed before consequence is created, and by preserving sufficient evidence to verify that determination independently.

## Section 5 — Authority-State Requirements

### 5.0 Purpose

This section defines the normative requirements for representing, evaluating, preserving, and verifying authority applicable to consequence-bearing digital actions.

Authority under the Trust-State Standard SHALL be represented as a verifiable state rather than inferred solely from identity, access, role, possession, or system capability.

This section is normative.

### 5.1 Authority-State Principle

A conforming system SHALL represent authority applicable to a proposed action in a manner sufficient to determine whether that authority is:

- valid;
- applicable;
- within scope;
- within temporal bounds;
- subject to relevant constraints;
- traceable to a defined source; and
- not revoked, suspended, superseded, or otherwise invalidated.

Authority SHALL be evaluated in relation to the specific action for which admissibility is being determined.

### 5.2 Authority-State

An Authority-State is a version-identifiable representation of the authority conditions applicable to an actor, action, resource, scope, or execution context at a defined point or interval in time.

An Authority-State SHALL contain or reference sufficient information to establish:

- authority source;
- authority subject;
- authority scope;
- applicable action or action class;

- applicable resource or resource class;
- temporal validity;
- delegation lineage, where applicable;
- constraints;
- revocation state;
- provenance;
- version or state identity; and
- integrity information sufficient to support verification.

An implementation MAY include additional attributes where required by the applicable Rule-State.

**5.3 Authority Source**

Each Authority-State SHALL identify the source from which authority originates.

The authority source SHALL be attributable to a defined:

- person;
- organization;
- system;
- governance body;
- legal instrument;
- contractual instrument;
- policy source;
- credential issuer;
- delegated authority source; or
- other objectively identifiable authority origin.

Authority SHALL NOT be treated as valid where its claimed source cannot be established under the applicable Rule-State.

**5.4 Authority Subject**

An Authority-State SHALL identify the actor, system, agent, role, process, or other subject to which authority applies.

Authority subject identity SHALL be sufficiently stable to prevent ambiguous or unintended substitution.

Where role-based authority is used, the system SHALL distinguish between:

- identity of the actor;
- role held by the actor; and
- authority derived from that role.

Possession of a role SHALL NOT establish authority outside the scope, constraints, and validity conditions associated with that role.

**5.5 Authority Scope**

Authority SHALL be bounded.

An Authority-State SHALL define the scope within which authority applies.

Scope MAY include:

- permitted action classes;
- prohibited action classes;
- applicable resources;
- applicable systems;
- transaction bounds;
- geographic or jurisdictional limits;
- organizational boundaries;
- temporal boundaries;
- dependency conditions;
- purpose limitations; and
- other constraints established by the governing source.

Authority SHALL NOT be interpreted as extending beyond its declared scope.

### 5.6 Action-Specific Authority

Authority sufficient for one action SHALL NOT be presumed sufficient for another action.

A conforming system SHALL determine whether the Authority-State applicable to the actor or system is sufficient for the specific proposed action.

Generalized identity assurance, access permission, account status, model authorization, application approval, or system role SHALL NOT by itself establish action-specific authority.

### 5.7 Authority Constraints

An Authority-State SHALL preserve constraints material to the exercise of authority.

Constraints MAY include:

- action limits;
- value limits;
- frequency limits;
- resource restrictions;
- approval conditions;
- dual-control requirements;
- separation-of-duty requirements;
- temporal restrictions;
- geographic restrictions;
- dependency requirements;
- contextual conditions;
- purpose restrictions; and
- escalation requirements.

A conforming system SHALL evaluate applicable authority constraints before determining authority sufficiency.

### 5.8 Authority Lineage

Where authority is derived, delegated, inherited, transferred, or otherwise dependent upon another authority source, a conforming system SHALL preserve sufficient lineage to establish the derivation path.

Authority lineage SHALL identify, where applicable:

- originating authority;
- delegating authority;
- receiving subject;
- delegation scope;
- delegation constraints;
- temporal validity;
- modifications;
- substitutions;
- supersession events; and
- revocation events.

A break in required authority lineage SHALL result in authority insufficiency unless the applicable Rule-State expressly permits another basis for authority.

**5.9 Delegated Authority**

Delegated authority SHALL NOT exceed the scope of the delegating authority.

A conforming system SHALL determine whether a delegation:

- originated from valid authority;
- remained within delegable scope;
- identified the receiving subject;
- preserved applicable constraints;
- remained within temporal bounds; and
- remained unrevoked at the time of evaluation.

Sub-delegation SHALL be permitted only where authorized by the applicable authority source and Rule-State.

### 5.10 Temporal Validity

Authority SHALL be evaluated at the time relevant to the proposed action.

An Authority-State SHALL identify temporal conditions necessary to determine whether authority was:

- active;
- not yet effective;
- expired;
- suspended;
- superseded;
- revoked; or
- otherwise invalid at the time of evaluation.

Prior authority SHALL NOT establish current authority where the governing authority state has materially changed.

### 5.11 Revocation and Suspension

A conforming system SHALL support determination of whether authority has been revoked, suspended, terminated, superseded, or otherwise invalidated.

Where revocation or suspension materially affects admissibility, the system SHALL:

- prevent continued reliance upon the invalid authority;
- identify the effective revocation state;
- preserve revocation provenance;
- preserve applicable lineage;
- re-evaluate dependent Authority-States where required; and
- re-evaluate affected admissibility determinations where required by the applicable Conformance Profile.

### 5.12 Authority-State Identity

Each Authority-State SHALL be version-identifiable or otherwise uniquely distinguishable from materially different authority conditions.

Authority-State identity SHALL change where a material change affects:

- authority source;
- authority subject;
- scope;
- constraints;
- delegation lineage;
- temporal validity;
- revocation status; or
- another condition material to admissibility.

Authority-State identity SHALL be integrity-protected.

**5.13 Authority-State Provenance**

A conforming system SHALL preserve provenance sufficient to establish the origin and relevant history of an Authority-State.

Provenance SHALL include, where applicable:

- origin;
- issuer or source;
- creation event;
- delegation events;
- modification events;
- validation events;
- suspension events;
- revocation events;
- supersession events; and
- relevant verification history.

Provenance SHALL be sufficient to support independent verification where required by the applicable Conformance Profile.

### 5.14 Authority Sufficiency

Authority sufficiency is the condition in which the applicable Authority-State establishes that the actor possesses the authority required for the proposed action within the applicable scope, constraints, context, and time.

Authority SHALL be considered sufficient only where the applicable Authority-State establishes, as required by the Rule-State:

- an attributable authority source;
- an identifiable authorized subject;
- authority applicable to the proposed action or action class;
- authority applicable to the affected resource or scope;
- satisfaction of applicable constraints;
- valid delegation lineage where authority is delegated;
- applicable temporal validity;
- absence of disqualifying suspension or revocation; and
- sufficient provenance and integrity to verify the authority relied upon.

Where authority sufficiency cannot be determined and established authority is required for admissibility, the authority state SHALL be treated as insufficient or indeterminate in accordance with the applicable Rule-State and Conformance Profile.

Possession of identity, access, credentials, role, or technical capability SHALL NOT by itself establish authority sufficiency.

### 5.15 Authority Insufficiency

Authority SHALL be considered insufficient where one or more required authority conditions cannot be established.

Authority insufficiency MAY result from:

- absent authority;
- unverifiable authority source;
- invalid delegation;
- broken authority lineage;
- scope mismatch;
- expired authority;
- suspended authority;
- revoked authority;
- unmet constraints;
- action mismatch;
- ambiguous authority subject; or
- inability to verify a condition required by the applicable Rule-State.

Where authority is required for admissibility, authority insufficiency SHALL prevent an admissible determination.

### 5.16 Authority-State Continuity

Authority-State validity SHALL NOT be presumed to persist following a material change.

A conforming system implementing continuity governance SHALL detect material changes affecting authority and determine whether previously valid authority remains sufficient.

Material authority change MAY include:

- delegation change;
- role change;
- scope change;
- constraint change;
- source change;

- temporal change;
- jurisdictional change;
- suspension;
- revocation; or
- supersession.

Where such change affects admissibility, dependent Trust-State and execution eligibility determinations SHALL be re-evaluated.

**5.17 Authority-State Serialization**

Where deterministic serialization is required by the applicable Conformance Profile, Authority-State data SHALL be represented in canonical form.

The canonical representation SHALL preserve the authority attributes necessary to support:

- identity;
- integrity verification;
- comparison;
- recomputation;
- replay; and
- independent verification.

Equivalent Authority-State inputs SHALL produce equivalent deterministic representations.

**5.18 Authority-State Integrity**

A conforming system SHALL protect Authority-State information against unauthorized modification.

Integrity controls SHALL be sufficient to detect material alteration to authority conditions relied upon in an admissibility determination.

Where an integrity value is used, the value SHALL be derived from the applicable canonical representation.

### 5.19 Independent Verification of Authority

Where required by the applicable Conformance Profile, a conforming system SHALL preserve sufficient information for an independent verifier to determine:

- the authority source;
- the authority subject;
- the applicable scope;
- applicable constraints;
- delegation lineage;
- temporal validity;
- revocation state;
- Authority-State identity;
- provenance; and
- whether authority was sufficient for the proposed action.

Independent verification SHALL NOT depend solely upon a declarative assertion that authority existed.

### 5.20 Authority and Identity Separation

Identity and authority SHALL be treated as distinct governance conditions.

Proof of identity MAY establish who or what an actor is.

Proof of authority SHALL establish whether that actor is permitted to perform the proposed action under the applicable governing conditions.

A verified identity SHALL NOT, by itself, constitute sufficient authority.

### 5.21 Authority and Access Separation

Access and authority SHALL be treated as distinct conditions.

A system MAY permit technical access to a resource without establishing authority to perform a specific consequence-bearing action involving that resource.

Possession of credentials, tokens, sessions, API access, administrative privileges, or technical capability SHALL NOT, by itself, establish action-specific authority.

### 5.22 Authority and Capability Separation

Technical capability SHALL NOT be treated as equivalent to authority.

The ability of a human, system, service, model, or autonomous agent to perform an action SHALL NOT establish that the action is admissible.

Conforming systems SHALL evaluate authority independently from capability where authority is required by the applicable Rule-State.

### 5.23 Autonomous and Machine Actors

Authority applicable to autonomous agents, artificial intelligence systems, software services, robotic systems, or other machine actors SHALL be represented and evaluated under the same principles of source, scope, constraints, lineage, temporal validity, and revocation.

Machine-generated intent, task assignment, model output, inferred objective, or technical capability SHALL NOT independently create authority.

Where a machine actor operates under delegated authority, the delegation SHALL be verifiable.

### 5.24 Conflicting Authority

Where multiple authority sources apply to a proposed action, the applicable Rule-State SHALL define how conflict is resolved.

A conforming system SHALL NOT resolve materially conflicting authority through arbitrary or undisclosed discretion.

Conflict resolution SHALL be objectively determinable and independently verifiable where required by the applicable Conformance Profile.

### 5.25 Unknown or Indeterminate Authority

Where required authority cannot be established with sufficient certainty under the applicable deterministic evaluation procedure, the authority state SHALL be treated as insufficient.

A conforming system SHALL NOT convert unknown, unavailable, ambiguous, or unverifiable authority into affirmative authority solely through inference.

**5.26 Authority-State Evidence**

A conforming system SHALL preserve evidence sufficient to establish the Authority-State relied upon in an admissibility determination.

Authority-State evidence SHALL include or reference the information necessary to establish:

- source;
- subject;
- scope;
- constraints;
- lineage;
- temporal validity;
- revocation state;
- version identity; and
- integrity.

**5.27 Conformance Requirement**

A system conforms to this section when it can demonstrate that authority relevant to consequence-bearing actions is represented, bounded, evaluated, preserved, and independently verifiable in accordance with the applicable Conformance Profile.

A generalized assertion that an actor, system, model, or service is “authorized” SHALL NOT constitute conformity.

**5.28 Canonical Statement**

Identity establishes who or what an actor is.

Access establishes what an actor can reach.

Capability establishes what an actor can do.

Authority establishes what an actor is permitted to do.

Admissibility determines whether that authority remains sufficient for this action, under these conditions, at this time.

**Section 6 — Evidence Requirements**

**6.0 Purpose**

This section defines the normative requirements for identifying, validating, preserving, evaluating, and independently verifying evidence used in admissibility determinations.

Evidence under the Trust-State Standard SHALL be treated as a governed input whose existence, validity, provenance, sufficiency, integrity, and applicability are independently determinable.

This section is normative.

**6.1 Evidence Principle**

A conforming system SHALL NOT treat the mere presence of information as sufficient evidence.

Evidence relied upon for admissibility SHALL be evaluated in relation to:

- the proposed action;
- the applicable Authority-State;
- the applicable Rule-State;
- the declared context;
- temporal validity;
- provenance;
- integrity;
- sufficiency requirements; and
- other governing conditions defined by the applicable Rule-State.

Evidence SHALL support a determination.

Evidence SHALL NOT replace the determination itself.

**6.2 Evidence**

Evidence is information, artifacts, records, credentials, attestations, observations, measurements, proofs, or other verifiable inputs used to support evaluation of admissibility.

Evidence MAY originate from:

- human actors;
- organizations;

- systems;
- sensors;
- credential issuers;
- trusted services;
- external authorities;
- cryptographic proofs;
- internal records;
- transaction records;
- policy systems; or
- other sources recognized by the applicable Rule-State.

**6.3 Evidence Requirement Identification**

Before admissibility is determined, a conforming system SHALL identify the evidence required by the applicable Rule-State.

The required evidence set SHALL be objectively determinable.

Evidence requirements MAY vary according to:

- action type;
- action significance;
- Authority-State;
- resource;
- jurisdiction;
- context;
- temporal conditions;
- risk classification;
- dependency conditions; and
- other Rule-State conditions.

A system SHALL NOT assert evidence sufficiency where required evidence has not been identified.

### 6.4 Evidence Existence

Evidence existence is the condition in which a required evidence item is present or retrievable for evaluation.

The existence of evidence SHALL NOT by itself establish:

- validity;
- authenticity;
- sufficiency;
- applicability;
- integrity;
- authority; or
- admissibility.

A conforming system SHALL distinguish evidence existence from evidence sufficiency.

### 6.5 Evidence Validity

Evidence validity is the condition in which evidence satisfies the validation requirements applicable to that evidence type.

Validation requirements MAY include:

- authenticity;
- issuer validity;
- source recognition;
- temporal validity;
- signature validity;
- credential status;
- schema validity;
- integrity verification;
- completeness;
- applicable assurance level; and
- other conditions defined by the Rule-State.

Invalid evidence SHALL NOT satisfy a requirement that expressly requires valid evidence.

### 6.6 Evidence Applicability

Evidence SHALL be evaluated for applicability to the proposed action.

Evidence applicable to one:

- actor;
- resource;
- action;
- transaction;
- context;
- jurisdiction;
- time period; or
- Rule-State

SHALL NOT be presumed applicable to another.

A conforming system SHALL prevent evidence from being applied outside its valid scope where scope is material to admissibility.

### 6.7 Evidence Sufficiency

Evidence sufficiency is the condition in which the required evidence set is present, valid, applicable, and adequate to satisfy the evidentiary requirements of the governing Rule-State.

A conforming system SHALL determine evidence sufficiency before asserting admissibility where evidence is required.

Evidence sufficiency SHALL be based upon objectively determinable conditions.

Evidence sufficiency SHALL NOT be inferred solely from:

- evidence volume;
- confidence score;
- model output;
- reputation;
- historical success;

- partial evidence;
- availability of substitute data; or
- generalized trust in the evidence source.

**6.8 Evidence Insufficiency**

Evidence SHALL be considered insufficient where one or more required evidence conditions cannot be established.

Evidence insufficiency MAY result from:

- missing evidence;
- invalid evidence;
- expired evidence;
- revoked evidence;
- unverifiable evidence;
- incomplete evidence;
- inapplicable evidence;
- insufficient provenance;
- failed integrity validation;
- ambiguous evidence scope;
- unmet assurance requirements; or
- inability to satisfy a Rule-State requirement.

Where sufficient evidence is required for admissibility, evidence insufficiency SHALL prevent an admissible determination.

**6.9 Evidence Provenance**

A conforming system SHALL preserve provenance sufficient to establish the origin and relevant history of evidence relied upon in an admissibility determination.

Evidence provenance SHALL identify, where applicable:

- evidence source;
- issuer;

- creator;
- collection event;
- creation time;
- transformation history;
- transmission history;
- validation history;
- storage history;
- relevant custody events;
- revocation or supersession events; and
- other material provenance information.

Provenance SHALL be sufficient to support independent verification where required by the applicable Conformance Profile.

**6.10 Evidence Lineage**

Where evidence is derived from, transformed from, aggregated from, or dependent upon other evidence, a conforming system SHALL preserve sufficient lineage to identify the relationship.

Evidence lineage SHALL identify, where applicable:

- source evidence;
- derived evidence;
- transformation process;
- aggregation process;
- validation process;
- dependency relationship; and
- material changes affecting evidentiary meaning.

Derived evidence SHALL NOT obscure the identity or status of material source evidence where source traceability is required.

### 6.11 Evidence Integrity

A conforming system SHALL protect evidence relied upon in admissibility determinations against unauthorized modification.

Integrity controls SHALL be sufficient to detect material alteration.

Evidence integrity MAY be established through:

- cryptographic hashing;
- digital signatures;
- authenticated data structures;
- tamper-evident storage;
- signed attestations;
- integrity-protected logs;
- deterministic integrity values; or
- other independently verifiable mechanisms.

Where an integrity value is used, it SHALL be derived from the applicable representation of the evidence.

### 6.12 Evidence Identity

Evidence relied upon in a determination SHALL be uniquely distinguishable or version-identifiable where material differences could affect admissibility.

Evidence identity SHALL change where a material modification changes the evidentiary content relied upon in a determination.

A conforming system SHALL preserve sufficient information to distinguish:

- original evidence;
- modified evidence;
- superseded evidence;
- revoked evidence; and
- replacement evidence.

### 6.13 Temporal Validity

Evidence SHALL be evaluated within the temporal conditions applicable to the proposed action.

A conforming system SHALL determine whether evidence is:

- current;
- expired;
- not yet valid;
- revoked;
- superseded;
- stale; or
- otherwise outside a valid temporal boundary

where such state is material to admissibility.

Historical validity SHALL NOT establish current validity where governing conditions require current evidence.

### 6.14 Evidence Revocation

Where evidence is revocable, a conforming system SHALL determine whether that evidence remained valid at the time relevant to the admissibility determination.

Where revocation materially affects admissibility, the system SHALL:

- prevent continued reliance upon revoked evidence;
- preserve revocation state;
- preserve revocation provenance;
- determine affected dependent evidence;
- re-evaluate evidence sufficiency where required; and
- re-evaluate dependent Trust-State or execution eligibility determinations where required by the applicable Conformance Profile.

### 6.15 Evidence Supersession

Evidence MAY become superseded without being revoked.

Where a newer evidence item materially replaces or changes the governing meaning of prior evidence, the system SHALL determine whether continued reliance upon the prior evidence remains permitted.

Superseded evidence SHALL remain distinguishable from current evidence.

**6.16 Conflicting Evidence**

Where materially conflicting evidence exists, the applicable Rule-State SHALL define how the conflict is resolved.

A conforming system SHALL NOT resolve material evidentiary conflict through arbitrary or undisclosed discretion.

Conflict resolution MAY include:

- source precedence;
- assurance precedence;
- temporal precedence;
- independent corroboration;
- authoritative-source rules;
- majority or quorum rules where expressly defined; or
- escalation to another governed decision process.

The resolution method SHALL be objectively determinable.

**6.17 Unknown or Indeterminate Evidence State**

Where required evidence cannot be validated, classified, or determined with sufficient certainty under the applicable evaluation procedure, the evidence state SHALL be treated as indeterminate.

Indeterminate evidence SHALL NOT be converted into affirmative evidentiary sufficiency solely through inference.

Where the Rule-State requires established evidence sufficiency, indeterminate evidence SHALL be treated as insufficient.

**6.18 Evidence Assurance**

The applicable Rule-State MAY require an evidence assurance level.

Evidence assurance MAY be based upon:

- source identity;
- source authority;
- issuance controls;
- verification method;
- cryptographic protections;
- attestation structure;
- collection controls;
- corroboration;
- chain of custody;
- validation method; or
- other objectively determinable assurance factors.

An assurance score or classification SHALL NOT replace the underlying evidence or its provenance.

**6.19 Attestations**

An attestation MAY constitute evidence where recognized by the applicable Rule-State.

An attestation SHALL identify or support identification of:

- the attesting party;
- the subject of the attestation;
- the asserted fact or condition;
- applicable scope;
- temporal validity;
- provenance;
- integrity; and
- any material constraints.

An attestation SHALL NOT be treated as independently sufficient merely because it is signed or issued by a recognized actor.

### 6.20 Credentials as Evidence

A credential MAY constitute evidence of identity, role, qualification, authority, status, entitlement, or another declared condition.

Possession of a credential SHALL NOT establish admissibility unless the credential satisfies the requirements applicable to the proposed action.

A conforming system SHALL distinguish:

- credential existence;
- credential validity;
- credential scope;
- credential status;
- credential authority; and
- evidentiary sufficiency.

### 6.21 Observational and Measured Evidence

Observations and measurements MAY constitute evidence where recognized by the Rule-State.

Where such evidence is relied upon, the system SHALL preserve sufficient information to establish, where applicable:

- measurement method;
- observation method;
- source;
- timestamp;
- calibration state;
- collection environment;
- transformation;
- validation;
- uncertainty bounds where material; and
- integrity.

A probabilistic or uncertain measurement MAY be used as evidence where expressly permitted by the Rule-State.

The governing determination SHALL remain subject to the deterministic requirements applicable to the declared Conformance Profile.

### 6.22 Machine-Generated Evidence

Evidence generated by an automated system, software service, artificial intelligence system, autonomous agent, or other machine process SHALL be subject to the same requirements for:

- provenance;
- identity;
- validity;
- scope;
- temporal status;
- integrity; and
- sufficiency

as evidence generated by human or organizational actors.

Machine generation SHALL NOT create a presumption of accuracy or authority.

### 6.23 Evidence from Probabilistic Systems

A probabilistic system MAY generate evidence or evidentiary inputs.

Where probabilistic evidence is used, the applicable Rule-State SHALL define:

- its admissible role;
- applicable thresholds;
- uncertainty treatment;
- required corroboration;
- deterministic bounding conditions; and
- whether the evidence may independently satisfy a requirement.

A probabilistic output SHALL NOT independently establish a deterministic fact unless the applicable Rule-State defines an objectively verifiable basis for doing so.

### 6.24 Evidence Preservation

A conforming system SHALL preserve sufficient evidence to support verification of determinations within the applicable retention boundary.

Evidence preservation SHALL maintain, where required:

- evidence identity;
- provenance;
- lineage;
- integrity;
- temporal state;
- validation state;
- revocation state;
- applicable scope; and
- relationship to the determination in which the evidence was used.

Preservation SHALL NOT require indefinite retention unless required by the applicable Rule-State, conformance specification, or external governing requirement.

### 6.25 Evidence Binding

Evidence relied upon in an admissibility determination SHALL be bound to that determination in a manner sufficient to establish which evidence was evaluated.

The binding SHALL prevent material ambiguity regarding the evidence set used to derive the determination.

Where canonical evidence representation is required, the determination SHALL reference or incorporate the applicable canonical evidence identity or integrity value.

### 6.26 Evidence and Authority Separation

Evidence and authority SHALL be treated as distinct governance conditions.

Evidence MAY establish facts relevant to authority.

Evidence SHALL NOT independently create authority unless the applicable Rule-State expressly recognizes the evidence source or instrument as authority-conferring.

Authority SHALL NOT eliminate the need for evidence where evidence is independently required.

### 6.27 Evidence and Rule-State Separation

Evidence and Rule-State SHALL be treated as distinct.

Evidence establishes or supports facts.

Rule-State establishes how facts are interpreted and what conditions must be satisfied.

A conforming system SHALL NOT permit evidence to silently alter the Rule-State under which it is evaluated.

### 6.28 Evidence and Admissibility Separation

Evidence sufficiency SHALL NOT be treated as equivalent to admissibility.

A proposed action MAY possess sufficient evidence while remaining non-admissible because of:

- insufficient authority;
- unmet Rule-State conditions;
- invalid context;
- revocation;
- dependency failure;
- policy constraint; or
- another governing condition.

Evidence is one input to admissibility.

### 6.29 Evidence Continuity

Evidence validity and sufficiency SHALL NOT be presumed to persist following material change.

A conforming system implementing continuity governance SHALL detect material evidence change where required and determine whether previously valid evidence remains sufficient.

Material change MAY include:

- expiration;
- revocation;
- supersession;
- modification;

- source invalidation;
- provenance change;
- integrity failure;
- scope change; or
- contextual change.

Where the change affects admissibility, dependent determinations SHALL be re-evaluated.

**6.30 Canonical Evidence Representation**

Where deterministic serialization is required by the applicable Conformance Profile, evidence relied upon in an admissibility determination SHALL be represented or referenced in canonical form sufficient to support deterministic evaluation.

The canonical representation SHALL preserve the material attributes necessary for:

- evidence identity;
- integrity verification;
- sufficiency determination;
- comparison;
- replay;
- recomputation; and
- independent verification.

Equivalent evidence inputs SHALL produce equivalent canonical representations under the applicable serialization procedure.

**6.31 Independent Verification of Evidence**

Where required by the applicable Conformance Profile, a conforming system SHALL preserve sufficient information for an independent verifier to determine:

- what evidence was required;
- what evidence was presented;
- the identity of that evidence;
- its provenance;

- its validity;
- its scope;
- its temporal state;
- its revocation or supersession state;
- its integrity;
- whether it was sufficient; and
- how it contributed to the admissibility determination.

Independent verification SHALL NOT depend solely upon a declarative assertion that required evidence existed.

**6.32 Evidence Recomputability**

Where independent recomputation is required, the system SHALL preserve or make reproducibly available the evidence necessary to recreate the evidentiary evaluation performed for the original determination.

Recomputation SHALL apply the equivalent:

- evidence set;
- evidence state;
- Rule-State;
- contextual conditions; and
- evaluation procedure

relevant to the original determination.

**6.33 Evidentiary State**

A conforming implementation MAY express the condition of evidence through defined evidentiary states.

Such states MAY include:

- PRESENT;
- VALID;
- SUFFICIENT;
- INSUFFICIENT;

- EXPIRED;
- REVOKED;
- SUPERSEDED;
- CONFLICTED;
- INDETERMINATE; and
- INVALID.

Where evidentiary states are used, their derivation criteria SHALL be objectively determinable and independently verifiable.

**6.34 Failure Handling**

Where required evidence cannot be established, validated, verified, or determined sufficient, the system SHALL behave in accordance with the applicable Rule-State and Conformance Profile.

Where evidence sufficiency is mandatory for admissibility, failure to establish sufficiency SHALL result in a non-admissible outcome.

A conforming system SHALL NOT silently substitute missing evidence with unsupported inference.

**6.35 Conformance Requirement**

A system conforms to this section when it can demonstrate that evidence relevant to consequence-bearing actions is:

- identified;
- validated;
- scoped;
- provenance-preserved;
- integrity-protected;
- evaluated for sufficiency;
- bound to the applicable determination; and
- independently verifiable

in accordance with the applicable Conformance Profile.

### 6.36 Canonical Statement

**Evidence is not merely information available to a system.**

Evidence is a governed input whose identity, provenance, validity, applicability, integrity, and sufficiency can be demonstrated.

Admissibility SHALL NOT be established from evidence that cannot be shown to satisfy the requirements applicable to the proposed action.

## Section 7 — Rule-State and Policy Identity

### 7.0 Purpose

This section defines the normative requirements for identifying, representing, preserving, evaluating, and independently verifying the rules and policy conditions governing admissibility determinations.

A conforming system SHALL preserve the identity and integrity of the Rule-State under which a determination is made.

This section is normative.

### 7.1 Rule-State Principle

Admissibility SHALL be determined under an identifiable Rule-State.

A Rule-State SHALL represent the version-identifiable set of rules, policies, constraints, governance conditions, and evaluation criteria applicable to a proposed action.

A conforming system SHALL NOT treat governing rules as implicit, unversioned, or indeterminate where those rules materially affect admissibility.

### 7.2 Rule-State

A Rule-State SHALL contain or reference sufficient information to establish:

- governing rule set;
- applicable policy set;
- rule identities;
- policy identities;
- rule versions;
- policy versions;
- applicable constraints;
- decision criteria;
- precedence rules;
- conflict-resolution rules;
- effective time;
- supersession state;

- provenance;
- integrity information; and
- other material governing conditions.

**7.3 Rule Identity**

Each material rule used in admissibility evaluation SHALL be uniquely identifiable.

Rule identity SHALL be sufficiently stable to distinguish:

- one rule from another;
- one version of a rule from another;
- active rules from superseded rules;
- amended rules from prior versions; and
- withdrawn or invalid rules from current rules.

A rule identifier SHALL NOT be reused for a materially different rule unless the versioning mechanism preserves distinguishability.

**7.4 Policy Identity**

Where a policy contains or governs one or more rules applicable to admissibility, the policy SHALL be identifiable.

Policy identity SHALL permit an independent verifier to determine:

- which policy applied;
- which version applied;
- when that policy became effective;
- whether it remained active at the time of evaluation; and
- whether it had been superseded, suspended, withdrawn, or modified.

**7.5 Rule-State Versioning**

A Rule-State SHALL be version-identifiable.

A new Rule-State version SHALL be created where a material change affects:

- applicable rules;
- applicable policies;
- rule content;
- rule precedence;
- constraint logic;
- decision criteria;
- conflict-resolution logic;
- temporal applicability;
- jurisdictional applicability; or
- other conditions material to admissibility.

Non-material editorial changes MAY be handled without creating a materially distinct Rule-State where the change does not alter evaluation behavior.

**7.6 Rule-State Provenance**

A conforming system SHALL preserve provenance sufficient to establish the origin and relevant history of the Rule-State.

Rule-State provenance SHALL include, where applicable:

- governance source;
- rule author or issuing body;
- creation event;
- approval event;
- effective date;
- modification history;
- supersession history;
- withdrawal history;
- validation history; and
- relevant governance authority.

### 7.7 Rule Applicability

A rule SHALL be evaluated only where applicable to the proposed action.

Rule applicability MAY depend upon:

- actor;
- action type;
- action class;
- resource;
- Authority-State;
- jurisdiction;
- context;
- temporal conditions;
- dependency conditions;
- risk class;
- policy domain; or
- other defined conditions.

A conforming system SHALL determine applicability before relying upon a rule in an admissibility determination.

### 7.8 Rule Sufficiency

Rule sufficiency is the condition in which the governing Rule-State contains sufficient rules and decision criteria to support the required admissibility determination.

A conforming system SHALL NOT assert deterministic admissibility where the governing Rule-State is materially incomplete, internally unresolved, or insufficient to determine the required outcome.

Where the Rule-State does not define a necessary condition, the system SHALL behave according to the applicable failure rule.

### 7.9 Rule-State Integrity

A conforming system SHALL protect Rule-State information against unauthorized modification.

Integrity controls SHALL be sufficient to detect material alteration to:

- rules;
- policy conditions;
- constraints;
- precedence logic;
- version identity;
- effective dates;
- revocation or supersession state; and
- other conditions relied upon in evaluation.

Where an integrity value is used, it SHALL be derived from the applicable canonical representation.

### 7.10 Rule-State Immutability for Historical Evaluation

A Rule-State relied upon in a completed determination SHALL remain reproducibly identifiable and SHALL NOT be materially altered after that determination is completed.

A conforming system SHALL preserve or make reproducibly available the Rule-State necessary to verify, recompute, or replay a prior determination.

Subsequent rule or policy changes SHALL be represented through a new Rule-State version or successor state.

A later Rule-State SHALL NOT silently replace, overwrite, or modify the Rule-State applicable to a prior determination.

### 7.11 Effective Time

A Rule-State SHALL identify the time or conditions under which it becomes effective.

A conforming system SHALL determine whether a Rule-State was:

- active;
- not yet effective;
- expired;
- suspended;
- superseded;
- withdrawn; or

- otherwise inactive

at the time relevant to the proposed action.

**7.12 Supersession**

Where a Rule-State is superseded, the system SHALL preserve the relationship between:

- superseded Rule-State;
- successor Rule-State;
- effective transition point; and
- material differences affecting admissibility.

Supersession SHALL NOT erase the identity or reproducibility of the prior Rule-State.

**7.13 Rule Withdrawal and Invalidation**

Where a rule or policy is withdrawn, invalidated, suspended, or otherwise rendered unusable, the system SHALL determine the effect upon:

- current admissibility determinations;
- pending execution eligibility;
- dependent Trust-States;
- continuity evaluation; and
- prior determinations where retrospective invalidation is expressly required.

**7.14 Rule Precedence**

Where multiple applicable rules differ in authority, specificity, jurisdiction, scope, or priority, the Rule-State SHALL define precedence.

Precedence MAY be based upon:

- governing authority;
- legal or regulatory hierarchy;
- policy hierarchy;
- specificity;

- temporal priority;
- jurisdiction;
- exception structure; or
- other objectively defined criteria.

A conforming system SHALL NOT resolve rule precedence through arbitrary or undisclosed discretion.

**7.15 Conflicting Rules**

Where materially conflicting rules apply, the Rule-State SHALL define how conflict is resolved.

Conflict resolution MAY include:

- precedence;
- deny-overrides;
- permit-overrides;
- most-specific-rule;
- authoritative-source precedence;
- quorum or multi-rule resolution;
- escalation; or
- another explicitly defined method.

The resolution method SHALL be objectively determinable and independently verifiable where required by the applicable Conformance Profile.

**7.16 Ambiguous Rules**

A materially ambiguous rule SHALL NOT be treated as deterministically resolved unless the Rule-State defines an applicable interpretation mechanism.

Where ambiguity cannot be resolved under the governing Rule-State, the affected condition SHALL be treated as indeterminate.

Where deterministic admissibility is required, unresolved material ambiguity SHALL prevent an admissible determination unless the Rule-State expressly defines another outcome.

### 7.17 Rule Evaluation Order

Where evaluation order affects outcome, the Rule-State SHALL define that order.

Evaluation order SHALL be deterministic.

A conforming system SHALL preserve sufficient information to identify the order in which material rules were evaluated where order affects the determination.

### 7.18 Conditional Rules

A conditional rule SHALL identify the condition under which it becomes applicable.

Conditional rules MAY depend upon:

- evidence state;
- Authority-State;
- context;
- temporal conditions;
- prior rule outputs;
- dependency state;
- jurisdiction;
- action classification; or
- other defined conditions.

Conditional rule activation SHALL be objectively determinable.

### 7.19 Exception Rules

Where exceptions are permitted, the Rule-State SHALL define:

- the applicable base rule;
- the exception condition;
- the authority capable of establishing the exception;
- the scope of the exception;
- temporal validity;
- evidence requirements;

- constraints; and
- precedence.

An exception SHALL NOT be inferred merely because a rule produces an undesirable or inconvenient outcome.

**7.20 Override Rules**

Where override authority exists, the Rule-State SHALL define:

- who or what may override;
- what may be overridden;
- applicable scope;
- required evidence;
- temporal conditions;
- accountability conditions; and
- preservation requirements.

An override SHALL itself be treated as a governed action.

**7.21 Human Judgment**

A Rule-State MAY permit human judgment where explicitly defined.

Where human judgment materially affects admissibility, the Rule-State SHALL define:

- when judgment is permitted;
- who may exercise it;
- applicable qualifications or authority;
- required evidence;
- decision boundaries;
- preservation requirements; and
- whether independent review is required.

Human discretion SHALL NOT be treated as deterministic unless its allowable decision space and resulting outputs are governed in a manner consistent with the applicable Conformance Profile.

### 7.22 Machine-Interpretable Rules

A Rule-State MAY be represented in machine-interpretable form.

Machine-readable or executable policy SHALL preserve:

- rule identity;
- policy identity;
- version;
- provenance;
- scope;
- constraints;
- precedence;
- effective state; and
- integrity.

Executable representation SHALL NOT alter the meaning of the governing rule without creating a materially distinct Rule-State.

### 7.23 Natural-Language Rules

Natural-language rules MAY form part of a Rule-State where permitted by the governing implementation.

Where natural-language interpretation affects admissibility, the interpretation mechanism SHALL be defined sufficiently to support the applicable Conformance Profile.

Where deterministic evaluation is required, unresolved semantic ambiguity SHALL NOT be silently converted into a deterministic outcome.

### 7.24 Probabilistic Rule Inputs

A Rule-State MAY incorporate probabilistic inputs, thresholds, classifications, or model outputs.

Where probabilistic inputs are used, the Rule-State SHALL define:

- the input source;
- threshold or decision boundary;
- uncertainty treatment;

- allowable role in evaluation;
- required corroboration;
- fallback behavior; and
- deterministic interpretation of the resulting condition.

Probabilistic input SHALL NOT eliminate the requirement for an identifiable Rule-State.

**7.25 Rule-State and Authority Separation**

Rule-State and Authority-State SHALL be treated as distinct.

Rule-State establishes the conditions governing whether authority is sufficient.

Authority-State establishes the authority conditions presented for evaluation.

A Rule-State SHALL NOT create factual authority merely by declaring that authority exists.

**7.26 Rule-State and Evidence Separation**

Rule-State and evidence SHALL be treated as distinct.

Rule-State establishes what evidence is required and how it is evaluated.

Evidence supplies the facts, artifacts, or proofs presented against those requirements.

Evidence SHALL NOT silently modify the Rule-State.

**7.27 Rule-State and Context Separation**

Rule-State and context SHALL be treated as distinct.

Rule-State defines how contextual conditions affect evaluation.

Context supplies the relevant state or conditions at the time of evaluation.

A change in context SHALL NOT alter the Rule-State unless the governing rules expressly make such change part of Rule-State selection.

**7.28 Rule-State Selection**

Where more than one Rule-State could govern a proposed action, the system SHALL deterministically select the applicable Rule-State or resolve the applicable combination under defined governance rules.

Rule-State selection SHALL be based upon objectively determinable conditions.

The selected Rule-State SHALL be preserved as part of the determination record.

### 7.29 Rule-State Binding

A completed admissibility determination SHALL be bound to the Rule-State under which it was derived.

The binding SHALL be sufficient to establish:

- Rule-State identity;
- version;
- integrity;
- effective status; and
- relationship to the determination.

### 7.30 Canonical Rule-State Representation

Where deterministic serialization is required by the applicable Conformance Profile, the Rule-State SHALL be represented in canonical form.

The canonical representation SHALL preserve material attributes necessary for:

- identity;
- version comparison;
- integrity verification;
- selection;
- replay;
- recomputation; and
- independent verification.

Equivalent Rule-State inputs SHALL produce equivalent canonical representations.

### 7.31 Independent Verification of Rule-State

Where required by the applicable Conformance Profile, a conforming system SHALL preserve sufficient information for an independent verifier to determine:

- which Rule-State applied;
- which rules and policies were applicable;
- which versions were used;
- the governing provenance;
- the effective state;
- material constraints;
- precedence and conflict-resolution logic;
- integrity; and
- whether the Rule-State was sufficient for the determination.

Independent verification SHALL NOT depend solely upon a declarative assertion that the correct policy was used.

**7.32 Rule-State Replay**

Where replay is required, the system SHALL preserve or make reproducibly available the Rule-State applicable to the original determination.

Replay SHALL apply the equivalent Rule-State version and equivalent evaluation logic relevant to the original determination.

**7.33 Rule-State Continuity**

A prior admissibility determination SHALL NOT be presumed valid following a material Rule-State change.

A conforming system implementing continuity governance SHALL detect material Rule-State change and determine whether dependent:

- Authority-State evaluations;
- evidence sufficiency determinations;
- Trust-States; and
- execution eligibility determinations

remain valid.

### 7.34 Governance Change Control

Material Rule-State changes SHALL be governed.

Change control SHALL preserve, where applicable:

- change origin;
- approving authority;
- change rationale;
- prior version;
- new version;
- effective time;
- impact boundary;
- affected rules;
- affected action classes;
- migration requirements; and
- verification history.

### 7.35 Failure Handling

Where the applicable Rule-State cannot be identified, verified, interpreted, or determined sufficient, the system SHALL behave according to the applicable Conformance Profile.

Where deterministic Rule-State evaluation is required, inability to establish the governing Rule-State SHALL prevent an admissible determination.

### 7.36 Conformance Requirement

A system conforms to this section when it can demonstrate that the Rule-State governing consequence-bearing actions is:

- identifiable;
- version-controlled;
- provenance-preserved;
- integrity-protected;
- applicable;

- sufficient;
- bound to the determination; and
- independently verifiable

in accordance with the applicable Conformance Profile.

**7.37 Cyclic and Transitive Rule Dependencies**

Where a Rule-State contains nested, imported, referenced, or transitive rule or policy dependencies, those dependencies SHALL remain sufficiently identifiable to support deterministic evaluation.

A conforming system SHALL detect a cyclic dependency where the cycle can:

- prevent completion of evaluation;
- alter Rule-State selection;
- alter rule applicability;
- alter precedence or conflict resolution;
- create unresolved dependency state; or
- otherwise affect the governed determination.

Where a cyclic dependency cannot be deterministically resolved under the applicable Rule-State or evaluation procedure, the affected condition SHALL be treated as indeterminate.

Where established Rule-State sufficiency is required for admissibility, an unresolved cyclic dependency SHALL prevent an admissible determination.

A conforming implementation SHALL NOT silently resolve a material cyclic dependency through implementation-specific recursion behavior, lookup order, caching behavior, or undocumented fallback logic.

**7.38 Canonical Statement**

Authority establishes whether an actor possesses permission.

Evidence establishes the facts relied upon in evaluation.

Rule-State establishes the governing conditions under which those facts and that authority are evaluated.

A determination cannot be independently reproduced if the rules that governed it cannot themselves be identified and reproduced.

## Section 8 — Admissibility Determination

### 8.0 Purpose

This section defines the normative requirements for determining whether a proposed consequence-bearing action, execution pathway, or state transition is admissible.

Admissibility determination is the governed evaluation point at which applicable Authority-State, evidence, Rule-State, context, and other governing conditions are evaluated before execution eligibility is established.

This section is normative.

### 8.1 Admissibility Principle

A conforming system SHALL determine admissibility before asserting execution eligibility where required by the applicable Conformance Profile.

Admissibility SHALL be derived from objectively determinable governing conditions.

An admissibility determination SHALL NOT be based solely upon:

- identity;
- access;
- technical capability;
- confidence;
- probability;
- reputation;
- prior authorization;
- historical behavior;
- system classification; or
- post-execution evidence.

Admissibility is action-specific and condition-specific.

### 8.2 Inputs to Admissibility

An admissibility determination SHALL evaluate the governing inputs applicable to the proposed action.

Applicable inputs SHALL include, where required by the Rule-State:

- Authority-State;
- evidence;
- Rule-State;
- context;
- temporal conditions;
- revocation state;
- dependency state;
- resource state;
- jurisdictional conditions;
- policy constraints; and
- other admissibility-governing conditions.

The determination SHALL identify or reference the material inputs upon which the outcome depends.

**8.3 Action Identification**

Each admissibility determination SHALL apply to an identifiable proposed action or action class.

The system SHALL preserve sufficient information to establish:

- the proposed action;
- the action subject;
- the relevant resource or target;
- the execution context;
- the applicable scope; and
- other action attributes required by the governing Rule-State.

A generalized assertion that an actor or system is authorized SHALL NOT substitute for identification of the proposed action.

**8.4 Evaluation Boundary**

The admissibility evaluation boundary SHALL define the conditions evaluated for the proposed action.

The boundary SHALL be sufficiently explicit to distinguish:

- included inputs;
- excluded inputs;
- authority scope;
- evidence scope;
- Rule-State scope;
- contextual scope;
- temporal scope; and
- dependency scope.

An admissibility claim SHALL NOT extend beyond its declared evaluation boundary.

**8.5 Authority Evaluation**

Where authority is required, the system SHALL determine whether the applicable Authority-State is sufficient for the proposed action.

Authority evaluation SHALL account for:

- authority source;
- subject;
- scope;
- constraints;
- lineage;
- temporal validity;
- revocation state; and
- action-specific applicability.

Authority insufficiency SHALL prevent an admissible determination where authority is required by the Rule-State.

**8.6 Evidence Evaluation**

Where evidence is required, the system SHALL determine whether the applicable evidence is sufficient for the proposed action.

Evidence evaluation SHALL account for:

- existence;
- validity;
- applicability;
- provenance;
- integrity;
- temporal validity;
- revocation or supersession;
- conflict; and
- sufficiency.

Evidence insufficiency SHALL prevent an admissible determination where sufficient evidence is required by the Rule-State.

**8.7 Rule-State Evaluation**

The system SHALL identify and evaluate the Rule-State governing the proposed action.

The Rule-State SHALL be:

- identifiable;
- version-identifiable;
- applicable;
- effective;
- integrity-protected; and
- sufficient to support the required determination.

Where the governing Rule-State cannot be established, deterministic admissibility SHALL NOT be asserted.

**8.8 Context Evaluation**

Where context affects admissibility, the system SHALL evaluate the contextual conditions applicable to the proposed action.

Context MAY include:

- time;

- location;
- jurisdiction;
- environment;
- actor state;
- resource state;
- system state;
- transaction state;
- dependency state; and
- other bounded conditions identified by the Rule-State.

Context SHALL be evaluated using objectively determinable inputs where required by the applicable Conformance Profile.

**8.9 Dependency Evaluation**

Where admissibility depends upon another state, system, approval, resource, action, or condition, the system SHALL evaluate the required dependency.

A dependency SHALL NOT be presumed satisfied solely because it was previously satisfied.

Where a material dependency cannot be established, the dependency state SHALL be treated according to the governing Rule-State.

**8.10 Temporal Evaluation**

Admissibility SHALL be evaluated within a defined temporal context where time is material.

The system SHALL account for temporal conditions affecting:

- Authority-State;
- evidence;
- Rule-State;
- context;
- dependencies;
- revocation;
- approvals; and

- other governing inputs.

A determination SHALL NOT rely upon stale governing conditions where the Rule-State requires current conditions.

**8.11 Revocation Evaluation**

The system SHALL evaluate applicable revocation states before asserting admissibility.

Revocation evaluation SHALL include, where applicable:

- authority revocation;
- credential revocation;
- evidence revocation;
- delegation revocation;
- Rule-State invalidation;
- approval withdrawal; and
- dependency invalidation.

A revoked governing input SHALL NOT be treated as valid unless the Rule-State expressly defines another treatment.

**8.12 Constraint Evaluation**

The system SHALL evaluate constraints applicable to the proposed action.

Constraints MAY include:

- action limitations;
- resource limitations;
- transaction limits;
- frequency limits;
- geographic restrictions;
- jurisdictional restrictions;
- temporal restrictions;
- purpose restrictions;

- dual-control requirements;
- approval conditions;
- separation-of-duty requirements; and
- other Rule-State constraints.

A material unmet constraint SHALL prevent an admissible determination where the Rule-State requires satisfaction.

**8.13 Deterministic Evaluation Procedure**

Where deterministic admissibility is required, the evaluation procedure SHALL be defined such that equivalent governing conditions and equivalent Rule-State versions produce equivalent outcomes.

The procedure SHALL define:

- applicable inputs;
- input normalization;
- rule selection;
- evaluation order where material;
- precedence;
- conflict resolution;
- treatment of indeterminate conditions;
- treatment of probabilistic inputs;
- failure behavior; and
- output derivation.

The procedure SHALL be independently reproducible where required by the applicable Conformance Profile.

**8.14 Evaluation Order**

Where the order of evaluation affects the outcome, the order SHALL be deterministically defined.

The system SHALL preserve sufficient information to establish the material evaluation sequence.

Evaluation order SHALL NOT depend upon undisclosed implementation discretion where such discretion can alter the determination.

**8.15 Intermediate Conditions**

A conforming system MAY produce intermediate evaluation conditions.

Intermediate conditions MAY include:

- AUTHORITY_SUFFICIENT;
- EVIDENCE_SUFFICIENT;
- RULE_STATE_VALID;
- CONTEXT_VALID;
- DEPENDENCIES_SATISFIED;
- REVOCATION_CLEAR;
- CONSTRAINTS_SATISFIED; and
- other implementation-defined states.

Where intermediate conditions affect the final determination, their derivation criteria SHALL be objectively determinable.

**8.16 Admissibility Outcome**

An admissibility determination SHALL produce a defined outcome.

At minimum, the implementation SHALL distinguish between:

- an outcome establishing that the applicable admissibility requirements are satisfied; and
- an outcome establishing that they are not satisfied.

An implementation MAY define additional states, including:

- ADMISSIBLE;
- CONDITIONALLY_ADMISSIBLE;
- NON_ADMISSIBLE;
- INDETERMINATE;
- DEGRADED; or
- other objectively defined classifications.

Classification criteria SHALL be defined by the applicable Rule-State or conformance specification.

**8.17 Conditional Admissibility**

A Rule-State MAY permit conditional admissibility.

Where conditional admissibility is used, the determination SHALL identify:

- the unmet or conditional requirement;
- the condition under which execution remains permitted;
- applicable constraints;
- temporal validity;
- required additional evidence;
- required monitoring or verification;
- expiration conditions; and
- conditions causing immediate loss of eligibility.

Conditional admissibility SHALL NOT be used to conceal an otherwise non-admissible state.

**8.18 Non-Admissibility**

A proposed action SHALL be classified as non-admissible where a governing requirement necessary for admissibility is not satisfied.

Non-admissibility MAY result from:

- authority insufficiency;
- evidence insufficiency;
- invalid Rule-State;
- unmet constraints;
- invalid context;
- dependency failure;
- revocation;
- material conflict;
- unresolved ambiguity;
- temporal invalidity; or
- another condition defined by the Rule-State.

A non-admissible outcome SHALL NOT be converted into execution eligibility except through a separately governed process authorized by the applicable Rule-State.

**8.19 Indeterminate Admissibility**

An admissibility state SHALL be treated as indeterminate where the system cannot establish whether required governing conditions are satisfied.

Indeterminate conditions MAY result from:

- unavailable inputs;
- unverifiable authority;
- unverifiable evidence;
- unresolved rule ambiguity;
- conflicting governing conditions;
- failed integrity checks;
- unavailable dependency state; or
- other unresolved material conditions.

Where the applicable Conformance Profile requires established admissibility before execution, an indeterminate state SHALL NOT result in execution eligibility.

**8.20 Failure Behavior**

The applicable Rule-State or Conformance Profile SHALL define failure behavior where admissibility cannot be established.

Where fail-closed behavior is required, the system SHALL prevent execution eligibility when:

- required inputs are unavailable;
- a material governing condition is indeterminate;
- integrity validation fails;
- authority cannot be established;
- evidence sufficiency cannot be established;
- Rule-State cannot be identified; or
- another required condition cannot be verified.

### 8.21 Determination Identity

Each material admissibility determination SHALL be uniquely identifiable or otherwise distinguishable.

Determination identity SHALL permit association with:

- action identity;
- Authority-State;
- evidence set;
- Rule-State;
- context;
- evaluation time;
- output;
- applicable Conformance Profile; and
- integrity information.

### 8.22 Determination Integrity

A conforming system SHALL protect the admissibility determination and its material inputs against unauthorized alteration.

Where an integrity value is used, it SHALL bind the determination to the applicable:

- action;
- Authority-State;
- evidence set;
- Rule-State;
- context;
- outcome; and
- other material governing inputs.

### 8.23 Determination Provenance

A conforming system SHALL preserve provenance sufficient to establish how an admissibility determination was produced.

Provenance SHALL include or reference, where applicable:

- initiating action request;
- evaluator identity;
- evaluation procedure;
- Authority-State;
- evidence;
- Rule-State;
- context;
- evaluation time;
- output; and
- subsequent continuity or revocation events.

**8.24 Determination Binding**

An admissibility determination SHALL be bound to the governing conditions under which it was derived.

The binding SHALL be sufficient to prevent material ambiguity regarding:

- what action was evaluated;
- what authority was evaluated;
- what evidence was evaluated;
- what rules governed;
- what context applied; and
- what outcome was produced.

**8.25 Trust-State Derivation**

Where a Trust-State is produced, it SHALL be derived from the admissibility determination and its governing conditions.

Trust-State SHALL represent the determined governance condition applicable to the declared action and evaluation boundary.

Trust-State SHALL NOT be treated as a generalized property of an actor, system, model, organization, or resource.

Equivalent governing conditions and equivalent Rule-State versions SHALL produce equivalent Trust-State outputs where deterministic derivation is required.

**8.26 Trust-State Classification**

A conforming system MAY classify Trust-State using defined states.

Such states MAY include:

- VALID;
- CONDITIONALLY_VALID;
- DEGRADED;
- REVOKED;
- NON_ADMISSIBLE;
- INDETERMINATE; or
- other objectively defined states.

Classification SHALL be derived from governing conditions and SHALL NOT be assigned through arbitrary or undisclosed discretion.

**8.27 Probabilistic Inputs**

Probabilistic inputs MAY contribute to admissibility evaluation where permitted by the Rule-State.

Where probabilistic inputs are used:

- their source SHALL be identified;
- their permitted role SHALL be defined;
- thresholds SHALL be explicit where applicable;
- uncertainty treatment SHALL be defined;
- required corroboration SHALL be identified; and
- the resulting governing condition SHALL be deterministically interpreted.

A probabilistic score alone SHALL NOT establish admissibility where the applicable Conformance Profile requires deterministic establishment of governing conditions.

**8.28 Human Decision Inputs**

Human decision or review MAY form part of an admissibility determination where permitted by the Rule-State.

Where human judgment materially affects the outcome, the system SHALL preserve sufficient information to establish:

- the authorized decision-maker;
- the authority under which judgment was exercised;
- the permissible decision boundary;
- evidence available to the reviewer;
- resulting decision; and
- applicable provenance.

Human intervention SHALL NOT bypass otherwise mandatory governance requirements unless the Rule-State expressly authorizes the intervention.

**8.29 Override and Exception Handling**

Overrides and exceptions affecting admissibility SHALL themselves be governed.

An override or exception SHALL identify:

- governing authority;
- applicable Rule-State;
- scope;
- justification;
- evidence;
- temporal validity;
- constraints;
- accountability conditions; and
- resulting determination.

An implementation SHALL NOT provide an undisclosed or ungoverned bypass around admissibility requirements.

### 8.30 Action-Level Determination

Admissibility SHALL be evaluated at the level necessary to govern the proposed consequence-bearing action.

System-level, model-level, application-level, role-level, credential-level, or account-level approval SHALL NOT by itself establish admissibility for a specific action.

The determination SHALL establish whether the proposed action is admissible under the governing conditions applicable at the time of evaluation.

### 8.31 Independent Verification

Where required by the applicable Conformance Profile, the system SHALL preserve sufficient information for an independent verifier to determine:

- which action was evaluated;
- which Authority-State applied;
- which evidence was required and used;
- which Rule-State governed;
- which contextual conditions applied;
- whether dependencies were satisfied;
- whether revocation was considered;
- how constraints were evaluated;
- which procedure was applied; and
- whether the resulting admissibility outcome was correctly derived.

Independent verification SHALL NOT depend solely upon a declarative assertion that the action was admissible.

### 8.32 Independent Recomputation

Where independent recomputation is required, a conforming system SHALL preserve or make reproducibly available the governing conditions and evaluation procedure necessary to recreate the original determination.

Equivalent:

- action inputs;

- Authority-State;
- evidence;
- Rule-State;
- context;
- temporal state;
- revocation state;
- dependency state; and
- evaluation procedure

SHALL produce an equivalent admissibility outcome.

**8.33 Replay**

Where replay is required, the system SHALL preserve sufficient information to reconstruct the determination under the governing conditions applicable at the original evaluation point.

Replay SHALL distinguish between:

- replay of the historical determination under historical conditions; and
- re-evaluation of the same proposed action under current conditions.

A current Rule-State SHALL NOT silently replace the historical Rule-State during replay.

**8.34 Determination Continuity**

An admissibility determination SHALL NOT be presumed valid following material change to governing conditions.

Where continuity governance is required, the system SHALL detect material change affecting:

- Authority-State;
- evidence;
- Rule-State;
- context;
- revocation;
- dependencies;

- constraints; or
- temporal validity.

The system SHALL determine whether the prior admissibility outcome remains valid.

**8.35 Determination Expiration**

A Rule-State MAY define an expiration period or validity window for an admissibility determination.

Where expiration is defined:

- the validity period SHALL be objectively determinable;
- expiration SHALL be detectable;
- expired determinations SHALL NOT establish current admissibility; and
- re-evaluation SHALL occur where required before execution.

**8.36 Determination Artifact**

A conforming implementation MAY generate a machine-verifiable artifact representing the admissibility determination.

Where such an artifact is generated, it SHOULD include or reference sufficient information to establish:

- determination identity;
- action identity;
- Trust-State;
- Authority-State identity;
- evidence identity;
- Rule-State identity;
- contextual boundary;
- evaluation time;
- outcome;
- applicable Conformance Profile; and
- integrity value.

Where required by the applicable Conformance Profile, the artifact SHALL support independent verification or recomputation.

### 8.37 Conformance Requirement

A system conforms to this section when it can demonstrate that admissibility determinations for consequence-bearing actions are:

- action-specific;
- based upon identified governing conditions;
- derived under an identifiable Rule-State;
- supported by sufficient authority and evidence where required;
- bounded to defined context;
- deterministically evaluated where required;
- integrity-protected;
- preserved with sufficient provenance; and
- independently verifiable or recomputable in accordance with the applicable Conformance Profile.

### 8.38 Evaluation-Path Equivalence

A conforming implementation MAY use short-circuit evaluation, optimized rule traversal, cached intermediate results, or another implementation-specific evaluation path.

Such behavior SHALL NOT alter the governed outcome that would result from application of the applicable Rule-State and governing conditions.

Where an implementation does not evaluate a branch, rule, or condition because its result cannot affect the governed outcome under the applicable deterministic procedure, the omission SHALL NOT constitute non-conformity.

Where evaluation-path information is material to independent verification, recomputation, replay, or conformance assessment, the implementation SHALL preserve sufficient information to establish why the path taken produced the required governed result.

Implementation-specific evaluation paths SHALL NOT conceal an unevaluated condition capable of materially altering the determination.

### 8.39 Canonical Statement

**Authority, evidence, Rule-State, and context do not independently authorize consequence.**

They are governed inputs to an admissibility determination.

Admissibility is the point at which those conditions converge to determine whether a specific consequence-bearing action may proceed.

## Section 9 — Execution Eligibility and Enforcement

### 9.0 Purpose

This section defines the normative requirements for deriving execution eligibility from admissibility and enforcing that determination before consequence-bearing execution is permitted to occur.

Execution eligibility is the governed transition from admissibility determination to permitted execution.

This section is normative.

### 9.1 Execution Eligibility Principle

A consequence-bearing action SHALL NOT be considered eligible for execution solely because:

- an actor is identified;
- access exists;
- capability exists;
- authority exists in general;
- evidence exists;
- a prior action was permitted;
- a model recommends execution; or
- the system is technically capable of proceeding.

Execution eligibility SHALL be derived from the admissibility determination applicable to the specific proposed action.

### 9.2 Relationship Between Admissibility and Execution Eligibility

Admissibility and execution eligibility SHALL be treated as distinct states.

Admissibility establishes whether the governing conditions applicable to a proposed action are satisfied.

Execution eligibility establishes whether that admissible action is permitted to proceed into consequence-bearing execution.

A conforming system SHALL NOT assert execution eligibility where required admissibility has not been established.

### 9.3 Execution Eligibility Determination

Where execution gating is required by the applicable Conformance Profile, a conforming system SHALL determine execution eligibility before consequence-bearing execution is permitted to form.

The determination SHALL account for:

- admissibility outcome;
- Trust-State;
- applicable Authority-State;
- Rule-State;
- contextual conditions;
- temporal validity;
- revocation state;
- dependency state;
- execution constraints; and
- other conditions identified by the Rule-State.

### 9.4 Eligibility Outcomes

A conforming implementation SHALL distinguish execution states sufficiently to prevent ambiguity regarding whether execution is permitted.

Execution eligibility states MAY include:

- ELIGIBLE;
- CONDITIONALLY_ELIGIBLE;
- NOT_ELIGIBLE;
- SUSPENDED;
- EXPIRED;
- REVOKED;
- INDETERMINATE; or
- other objectively defined states.

The derivation criteria for each state SHALL be objectively determinable.

**9.5 Eligible State**

An action MAY be classified as ELIGIBLE only where all governing conditions required for execution have been satisfied.

An ELIGIBLE state SHALL be bounded to the:

- proposed action;
- actor or executing subject;
- resource or target;
- applicable Authority-State;
- evidence set;
- Rule-State;
- context;
- temporal conditions; and
- other material execution conditions.

Eligibility SHALL NOT be treated as a generalized or indefinite permission.

**9.6 Conditional Eligibility**

A Rule-State MAY permit conditional execution eligibility.

Where conditional eligibility is permitted, the determination SHALL identify:

- the applicable condition;
- remaining constraint;
- required additional control;
- applicable scope;
- temporal validity;
- monitoring or verification requirement;
- revocation condition; and
- condition causing eligibility to terminate.

Conditional eligibility SHALL NOT be used to bypass a requirement that the Rule-State designates as mandatory before execution.

### 9.7 Non-Eligibility

A proposed action SHALL be classified as not eligible for execution where:

- admissibility is not established;
- required authority is insufficient;
- required evidence is insufficient;
- the Rule-State is invalid or unavailable;
- a required constraint is unsatisfied;
- applicable context is invalid;
- a dependency has failed;
- revocation applies;
- temporal validity has expired;
- an integrity failure exists; or
- another mandatory execution condition is unsatisfied.

### 9.8 Indeterminate Eligibility

Where execution eligibility cannot be established because a material governing condition is unknown, unavailable, unverifiable, or unresolved, the eligibility state SHALL be treated as indeterminate.

Where the applicable Conformance Profile requires established eligibility before execution, an indeterminate state SHALL NOT permit consequence-bearing execution.

### 9.9 Execution Boundary

A conforming system implementing execution enforcement SHALL define the execution boundary at which a proposed action transitions from governed evaluation into consequence-bearing execution.

The execution boundary SHALL be sufficiently explicit to determine:

- when eligibility evaluation is complete;
- when enforcement occurs;
- when consequence-bearing execution begins; and
- whether execution occurred without valid eligibility.

### 9.10 Pre-Execution Enforcement

Where required by the applicable Conformance Profile, enforcement SHALL occur before consequence-bearing execution is permitted to form.

A conforming system SHALL:

- evaluate execution eligibility;
- enforce the resulting eligibility state;
- prevent execution when eligibility is absent;
- prevent execution when eligibility is indeterminate where fail-closed behavior is required; and
- preserve evidence of the enforcement decision.

### 9.11 Fail-Closed Behavior

Where fail-closed behavior is required, the system SHALL deny or prevent execution if:

- admissibility cannot be established;
- execution eligibility cannot be established;
- required governing inputs are unavailable;
- integrity validation fails;
- required authority cannot be verified;
- required evidence cannot be verified;
- Rule-State identity cannot be established;
- revocation status cannot be resolved where resolution is mandatory; or
- another required execution condition cannot be verified.

Failure to evaluate SHALL NOT be treated as permission to execute.

### 9.12 Enforcement Mechanism

A conforming implementation MAY use any enforcement mechanism capable of reliably preventing non-eligible execution.

Enforcement mechanisms MAY include:

- execution gates;

- policy enforcement points;
- transaction controls;
- API gateways;
- workflow controls;
- command interceptors;
- state-transition guards;
- authorization middleware;
- hardware controls;
- execution brokers; or
- other mechanisms satisfying the normative requirements of this section.

The Standard does not require a specific enforcement technology.

**9.13 Enforcement Binding**

The enforcement event SHALL be bound to the execution eligibility determination upon which it relies.

The binding SHALL be sufficient to establish:

- which action was evaluated;
- which eligibility state applied;
- which admissibility determination supported eligibility;
- which Rule-State governed;
- which Authority-State applied;
- which material evidence was relied upon; and
- whether execution was permitted or prevented.

**9.14 Enforcement Timing**

Execution eligibility SHALL be evaluated sufficiently close to the execution boundary to ensure that governing conditions remain valid at the time of execution.

Where material change may occur between determination and execution, the applicable Rule-State or Conformance Profile SHALL define whether:

- eligibility remains valid;
- revalidation is required;
- re-evaluation is required; or
- execution must be suspended.

**9.15 Eligibility Validity Window**

A Rule-State MAY define a temporal validity window for execution eligibility.

Where such a window exists:

- the start condition SHALL be identifiable;
- the expiration condition SHALL be identifiable;
- the validity window SHALL be objectively determinable;
- expired eligibility SHALL NOT permit execution; and
- re-evaluation SHALL occur where required.

**9.16 One-Time and Reusable Eligibility**

Execution eligibility MAY be:

- single-use;
- reusable for a defined number of executions;
- reusable during a defined time period;
- reusable within a bounded transaction;
- reusable within a defined workflow; or
- otherwise constrained by the Rule-State.

Reusable eligibility SHALL preserve sufficient scope controls to prevent use outside the authorized execution boundary.

**9.17 Execution Scope**

Execution eligibility SHALL apply only within its declared scope.

Scope MAY include:

- action;
- action class;
- resource;
- resource class;
- transaction;
- actor;
- system;
- geographic boundary;
- jurisdiction;
- time period;
- workflow;
- purpose; or
- other Rule-State conditions.

A conforming system SHALL prevent execution eligibility from being extended beyond its governed scope.

**9.18 Eligibility and Capability Separation**

Technical capability and execution eligibility SHALL remain distinct.

A system MAY possess the technical capability to execute an action while the action remains not eligible.

Technical capability SHALL NOT override a non-eligible state.

**9.19 Eligibility and Access Separation**

Access and execution eligibility SHALL remain distinct.

Possession of access to a system or resource SHALL NOT, by itself, permit a consequence-bearing action.

A conforming system SHALL prevent access rights from silently functioning as action-level execution authorization where the Rule-State requires separate eligibility determination.

### 9.20 Eligibility and Authority Separation

Authority and execution eligibility SHALL remain distinct.

Valid authority MAY be necessary for eligibility but SHALL NOT be sufficient where additional evidence, Rule-State, contextual, dependency, temporal, or other conditions remain unsatisfied.

### 9.21 Eligibility and Trust-State

Where a Trust-State is used to support execution eligibility, the eligibility determination SHALL identify the Trust-State upon which execution depends.

The Trust-State SHALL be valid for the action and governing conditions at the time relevant to execution.

A stale, expired, degraded, revoked, or otherwise invalid Trust-State SHALL NOT establish eligibility where the Rule-State requires a valid Trust-State.

### 9.22 Revocation Before Execution

A conforming system SHALL account for revocation occurring before execution.

Where revocation materially invalidates:

- authority;
- evidence;
- delegation;
- approval;
- Rule-State;
- dependency; or
- Trust-State,

the associated execution eligibility SHALL be invalidated or re-evaluated as required by the Rule-State.

### 9.23 Revocation During Pending Execution

Where an action remains pending between eligibility determination and consequence-bearing execution, the system SHALL define how material revocation is handled.

The Rule-State MAY require:

- immediate cancellation;

- suspension;
- re-evaluation;
- reauthorization; or
- another objectively defined response.

**9.24 Material Change Before Execution**

Execution eligibility SHALL NOT be presumed valid following material change to governing conditions.

Material change MAY include:

- Authority-State change;
- evidence change;
- Rule-State change;
- context change;
- dependency change;
- resource state change;
- temporal change;
- revocation; or
- another condition material to eligibility.

Where required, the system SHALL re-evaluate eligibility before execution.

**9.25 Execution After Material Change**

Where execution occurs after a material change but before required re-evaluation, the implementation SHALL treat the event as an enforcement failure unless the Rule-State expressly permitted continued eligibility.

Such failures SHALL be detectable and preserved for conformity assessment.

**9.26 Multi-Step Execution**

Where a consequence-bearing action consists of multiple material execution steps, the Rule-State SHALL define whether eligibility applies:

- to the entire sequence;
- to individual steps;
- to defined checkpoints; or
- to another bounded execution unit.

A system SHALL re-evaluate eligibility at intermediate steps where required by the governing conditions.

**9.27 Chained Actions**

Where one action initiates or permits another consequence-bearing action, each action SHALL be evaluated at the level required by the applicable Rule-State.

Eligibility for an initiating action SHALL NOT automatically establish eligibility for all downstream actions.

**9.28 Autonomous Execution**

An autonomous agent or system SHALL NOT infer continuing execution eligibility solely from:

- initial task assignment;
- goal specification;
- system role;
- prior approval;
- available credentials;
- technical capability; or
- prior successful execution.

Where consequence-bearing actions occur autonomously, action-level eligibility SHALL be established according to the applicable Conformance Profile and Rule-State.

**9.29 Human-Initiated Execution**

A human instruction or approval MAY form part of execution eligibility where recognized by the Rule-State.

Human initiation SHALL NOT bypass:

- Authority-State requirements;
- evidence requirements;

- Rule-State requirements;
- contextual requirements; or
- other mandatory admissibility conditions.

**9.30 Emergency Execution**

A Rule-State MAY define an emergency execution pathway.

Where such a pathway exists, it SHALL define:

- emergency qualification conditions;
- authorized actors;
- applicable authority;
- evidence requirements;
- scope;
- temporal validity;
- override conditions;
- accountability requirements;
- preservation requirements; and
- post-event review requirements.

Emergency execution SHALL remain governed execution.

**9.31 Override of Execution Eligibility**

An override of a non-eligible state SHALL be permitted only where expressly authorized by the Rule-State.

An override SHALL identify:

- override authority;
- justification;
- applicable evidence;
- scope;
- constraints;

- temporal validity;
- accountability;
- resulting eligibility state; and
- provenance.

The override itself SHALL be independently verifiable where required by the applicable Conformance Profile.

**9.32 Enforcement Evidence**

A conforming system SHALL preserve evidence sufficient to establish:

- the execution eligibility state;
- the action to which it applied;
- the enforcement mechanism;
- the enforcement time;
- whether execution was permitted or prevented;
- applicable Rule-State;
- associated admissibility determination; and
- relevant integrity information.

**9.33 Execution Event Binding**

Where execution occurs, the execution event SHALL be bound to the eligibility determination that permitted it where required by the applicable Conformance Profile.

The binding SHALL permit an independent verifier to determine whether the action executed under valid eligibility.

**9.34 Execution Without Eligibility**

A conforming system implementing enforcement SHALL detect execution that occurs:

- without an eligibility determination;
- under expired eligibility;
- under revoked eligibility;

- outside eligibility scope;
- under a mismatched action;
- under an invalid Rule-State; or
- after a material change requiring re-evaluation.

Such execution SHALL constitute an enforcement failure unless expressly permitted by the governing Rule-State.

**9.35 Duplicate and Replay Execution**

Where duplicate execution or replay of an action may create consequence, the Rule-State SHALL define how such execution is governed.

Controls MAY include:

- single-use eligibility;
- nonce or transaction identity;
- sequence controls;
- idempotency requirements;
- replay prevention; or
- other deterministic safeguards.

A valid prior eligibility determination SHALL NOT automatically authorize an unintended duplicate execution.

**9.36 Execution Integrity**

A conforming system SHALL protect the relationship between:

- action request;
- admissibility determination;
- Trust-State;
- execution eligibility;
- enforcement decision; and
- resulting execution event

against unauthorized substitution or alteration.

### 9.37 Independent Verification of Eligibility

Where required by the applicable Conformance Profile, a conforming system SHALL preserve sufficient information for an independent verifier to determine:

- whether admissibility was established;
- which eligibility state was produced;
- whether that state was valid at execution time;
- whether the action remained within scope;
- whether revocation or material change occurred;
- whether enforcement behaved correctly; and
- whether the execution event corresponded to the permitted action.

### 9.38 Independent Recomputation of Eligibility

Where independent recomputation is required, a conforming system SHALL preserve or make reproducibly available the governing inputs and procedure necessary to recreate the execution eligibility determination.

Equivalent governing conditions SHALL produce an equivalent eligibility outcome under the applicable deterministic procedure.

### 9.39 Enforcement Replay

Where replay is required, the system SHALL permit reconstruction of whether the enforcement mechanism should have permitted or prevented the action under the governing conditions applicable at the original execution boundary.

Replay SHALL distinguish historical verification from current re-evaluation.

### 9.40 Enforcement Failure Classification

A conforming implementation MAY classify enforcement failures.

Failure classes MAY include:

- NO_DETERMINATION;
- INVALID_ELIGIBILITY;
- EXPIRED_ELIGIBILITY;

- REVOKED_ELIGIBILITY;
- SCOPE_VIOLATION;
- ACTION_MISMATCH;
- MATERIAL_CHANGE_NOT_REEVALUATED;
- GATE_BYPASS;
- INTEGRITY_FAILURE;
- DUPLICATE_EXECUTION; and
- other objectively defined classifications.

Where failure classes are used, their derivation criteria SHALL be documented and independently verifiable.

**9.41 Conformance Requirement**

A system conforms to this section when it can demonstrate that execution eligibility for consequence-bearing actions is:

- derived from admissibility;
- action-specific;
- bounded to defined governing conditions;
- evaluated before execution where required;
- enforced at the execution boundary;
- invalidated or re-evaluated following material change where required;
- integrity-protected;
- bound to resulting execution events where required; and
- independently verifiable or recomputable in accordance with the applicable Conformance Profile.

**9.42 Canonical Statement**

**Admissibility answers whether an action satisfies the conditions necessary to proceed.**

Execution eligibility answers whether that admissible action may cross the execution boundary now.

Enforcement ensures that an action that is not eligible does not become consequence.

## Section 10 — Trust-State Continuity and Revocation

### 10.0 Purpose

This section defines the normative requirements for maintaining, re-evaluating, degrading, suspending, revoking, or terminating Trust-State validity as governing conditions change over time.

Trust-State SHALL NOT be presumed permanent.

A Trust-State is valid only while the governing conditions upon which it depends remain sufficient under the applicable Rule-State.

This section is normative.

### 10.1 Continuity Principle

A conforming system SHALL determine whether a previously derived Trust-State remains valid following material change to the conditions upon which that Trust-State depends.

Trust-State continuity SHALL be governed through objectively determinable conditions.

A prior valid determination SHALL NOT establish indefinite future validity.

### 10.2 Trust-State Continuity

Trust-State continuity is the condition in which a previously derived Trust-State:

- remains valid because material governing conditions remain sufficient; or
- is deterministically re-evaluated when material governing conditions change.

Continuity SHALL be evaluated in accordance with the applicable Conformance Profile.

### 10.3 Continuity Boundary

A Trust-State SHALL have a defined continuity boundary.

The continuity boundary SHALL identify the conditions material to continued validity, including where applicable:

- Authority-State;
- evidence;
- Rule-State;
- context;

- temporal validity;
- dependency state;
- revocation state;
- execution scope;
- resource state;
- actor state; and
- other conditions defined by the Rule-State.

A continuity claim SHALL NOT extend beyond the declared continuity boundary.

**10.4 Material Change**

A material change is an objectively determinable alteration to a governing condition that may affect the validity of a prior Trust-State, admissibility determination, or execution eligibility determination.

Material change MAY include:

- Authority-State modification;
- evidence modification;
- evidence expiration;
- evidence revocation;
- Rule-State modification;
- Rule-State supersession;
- context change;
- dependency change;
- jurisdictional change;
- temporal expiration;
- resource-state change;
- actor-state change;
- execution-condition change;
- suspension;
- revocation; or

- integrity failure.

**10.5 Materiality Determination**

The applicable Rule-State SHALL define or support determination of which changes are material.

A change SHALL be considered material where it can alter:

- authority sufficiency;
- evidence sufficiency;
- Rule-State applicability;
- contextual validity;
- admissibility;
- Trust-State classification;
- execution eligibility; or
- another condition required for continued validity.

A conforming system SHALL NOT require re-evaluation for changes that cannot affect the applicable determination unless the Rule-State expressly requires it.

**10.6 Change Detection**

Where continuity governance is required, a conforming system SHALL detect material change within the applicable continuity boundary.

Detection mechanisms MAY include:

- event notification;
- state comparison;
- revocation feeds;
- version comparison;
- expiration checks;
- integrity validation;
- dependency monitoring;
- authoritative-source updates;

- periodic validation; or
- other deterministic mechanisms.

Change detection SHALL be sufficiently timely to satisfy the applicable Rule-State.

**10.7 Continuity Evaluation**

Upon detection of material change, the system SHALL determine whether the prior Trust-State remains valid.

Continuity evaluation SHALL consider, where applicable:

- the prior determination;
- changed governing condition;
- unaffected governing conditions;
- applicable Rule-State;
- temporal state;
- revocation state;
- dependency state; and
- execution status.

**10.8 Re-Evaluation Requirement**

Where material change may affect admissibility or execution eligibility, the system SHALL re-evaluate the affected determination when required by the applicable Conformance Profile or Rule-State.

Re-evaluation SHALL apply the governing conditions relevant at the time of re-evaluation.

A re-evaluation is distinct from historical replay.

**10.9 Historical Replay and Current Re-Evaluation**

A conforming system SHALL distinguish between:

- historical replay of a prior determination under the conditions applicable at the original evaluation time; and
- current re-evaluation of the same or related action under current governing conditions.

Historical replay SHALL preserve the original Rule-State and governing inputs.

Current re-evaluation SHALL apply the Rule-State and governing conditions applicable to the current evaluation point.

**10.10 Trust-State Validity**

A Trust-State SHALL remain valid only while the conditions necessary to support its classification remain satisfied.

Validity MAY depend upon:

- continued Authority-State sufficiency;
- continued evidence sufficiency;
- continued Rule-State applicability;
- continued contextual validity;
- continued dependency satisfaction;
- absence of material revocation;
- absence of material integrity failure; and
- continued satisfaction of temporal conditions.

**10.11 Trust-State Expiration**

A Rule-State MAY define a temporal validity period for a Trust-State.

Where expiration applies:

- the validity period SHALL be objectively determinable;
- the expiration point SHALL be detectable;
- expired Trust-State SHALL NOT establish current admissibility or execution eligibility; and
- re-evaluation SHALL occur where required before further execution.

**10.12 Trust-State Degradation**

A Trust-State MAY be classified as degraded where governing conditions remain partially valid but one or more conditions have materially weakened without requiring immediate revocation.

Where DEGRADED is used, the Rule-State SHALL define:

- degradation criteria;

- permitted actions;
- prohibited actions;
- remediation requirements;
- re-evaluation requirements;
- temporal limits; and
- escalation conditions.

A degraded Trust-State SHALL NOT be treated as equivalent to a fully valid Trust-State.

**10.13 Trust-State Suspension**

A Trust-State MAY be suspended where continued validity cannot be established but permanent invalidation has not yet been determined.

Where suspension is used:

- the suspension condition SHALL be identifiable;
- execution consequences SHALL be defined;
- required investigation or remediation SHALL be defined;
- restoration criteria SHALL be defined; and
- suspension provenance SHALL be preserved.

Where the Rule-State requires valid Trust-State before execution, a suspended Trust-State SHALL NOT support execution eligibility.

**10.14 Trust-State Revocation**

Trust-State revocation is the governed invalidation of a previously derived Trust-State.

A Trust-State SHALL be revoked where the applicable Rule-State requires revocation due to a material condition such as:

- revoked authority;
- invalidated evidence;
- revoked credential;
- invalid delegation;
- material Rule-State invalidation;

- integrity failure;
- prohibited contextual condition;
- disqualifying dependency change;
- material fraud or falsification; or
- another defined revocation condition.

**10.15 Revocation Principle**

Revocation SHALL be treated as a state transition, not merely as an annotation.

Where revocation affects continued admissibility or execution eligibility, the system SHALL prevent further reliance upon the revoked state.

**10.16 Revocation Source**

A revocation event SHALL identify or support identification of the authority or condition capable of producing revocation.

The revocation source SHALL be verifiable where required by the Rule-State.

A revocation SHALL NOT be accepted solely because an unverified actor or system asserts that revocation occurred.

**10.17 Revocation Scope**

A revocation SHALL define its scope.

Scope MAY include:

- specific Trust-State;
- Authority-State;
- credential;
- evidence item;
- rule;
- delegation;
- action class;
- resource;

- actor;
- transaction;
- time period; or
- other governed boundary.

A revocation SHALL NOT be applied beyond its valid scope unless required by the Rule-State.

**10.18 Revocation Effective Time**

A revocation event SHALL identify the effective time or effective condition of revocation where time is material.

The system SHALL distinguish between:

- revocation decision time;
- effective revocation time;
- detection time; and
- enforcement time

where these differ materially.

**10.19 Revocation Lineage**

A conforming system SHALL preserve sufficient lineage to determine the relationship between a revocation and the state or governing input affected.

Revocation lineage SHALL identify, where applicable:

- revocation source;
- affected state;
- affected version;
- effective time;
- reason or governing condition;
- dependent states;
- subsequent remediation; and
- restoration or replacement state.

**10.20 Revocation Propagation**

Where a revoked governing input affects dependent determinations, the applicable Rule-State SHALL define whether revocation propagates.

Propagation MAY affect:

- derived Authority-States;
- evidence sufficiency;
- dependent Trust-States;
- execution eligibility;
- pending actions;
- reusable eligibility;
- delegated authority; and
- downstream state transitions.

Propagation SHALL be objectively determinable.

**10.21 Authority Revocation**

Where authority relied upon in a Trust-State is revoked, the system SHALL determine whether the affected Trust-State remains valid.

Where the revoked authority was necessary for admissibility:

- the Trust-State SHALL be invalidated, degraded, suspended, or re-evaluated as defined by the Rule-State;
- dependent execution eligibility SHALL be invalidated where required; and
- further execution SHALL be prevented where required by the applicable Conformance Profile.

**10.22 Evidence Revocation**

Where evidence relied upon in a Trust-State is revoked or otherwise invalidated, the system SHALL determine whether sufficient evidence remains.

Where evidence sufficiency no longer exists, the dependent Trust-State SHALL NOT remain fully valid unless the Rule-State permits another evidentiary basis.

**10.23 Rule-State Change**

A material Rule-State change MAY require re-evaluation of existing Trust-States.

The applicable Rule-State governance process SHALL define whether a change applies:

- prospectively only;
- to pending actions;
- to reusable execution eligibility;
- to active Trust-States; or
- retrospectively.

A later Rule-State SHALL NOT automatically invalidate a historical determination unless the governing change rules expressly require such effect.

**10.24 Context Change**

Where context is material to admissibility, a material context change SHALL trigger continuity evaluation where required.

Context change MAY include:

- geographic movement;
- jurisdictional change;
- system-state change;
- resource-state change;
- transaction-state change;
- environmental change;
- actor-state change; or
- other conditions identified by the Rule-State.

**10.25 Dependency Change**

Where a Trust-State depends upon another state or system condition, loss or degradation of that dependency SHALL trigger continuity evaluation where material.

A dependency previously satisfied SHALL NOT be presumed permanently satisfied.

**10.26 Integrity Failure**

A material integrity failure affecting a governing input or determination SHALL trigger continuity evaluation.

Integrity failure MAY require:

- suspension;
- invalidation;
- revocation;
- evidence replacement;
- re-verification;
- re-evaluation; or
- other Rule-State-defined action.

A state whose integrity cannot be established SHALL NOT be presumed valid where integrity is required.

**10.27 Pending Execution**

Where execution eligibility exists but execution has not yet occurred, material change affecting the underlying Trust-State SHALL be evaluated before execution where required.

The system SHALL prevent execution under stale eligibility where re-evaluation is required.

**10.28 Active or Multi-Step Execution**

Where consequence-bearing execution occurs across multiple steps, the Rule-State SHALL define whether continuity must be maintained throughout execution.

Continuity controls MAY require:

- checkpoint evaluation;
- revalidation before critical steps;
- continuous revocation awareness;
- suspension on material change;
- termination; or
- controlled rollback.

**10.29 Long-Running Autonomous Activity**

Where an autonomous system operates over an extended period, initial admissibility SHALL NOT establish indefinite authority to continue consequence-bearing action.

The applicable Rule-State SHALL define:

- continuity checkpoints;
- validity windows;
- re-evaluation triggers;
- revocation handling;
- authority refresh conditions;
- evidence refresh conditions; and
- termination conditions.

**10.30 Continuity State Classification**

A conforming implementation MAY classify continuity state.

Such classifications MAY include:

- CONTINUING;
- REEVALUATION_REQUIRED;
- DEGRADED;
- SUSPENDED;
- EXPIRED;
- REVOKED;
- TERMINATED;
- INDETERMINATE; or
- other objectively defined states.

Where continuity states are used, derivation criteria SHALL be objectively determinable.

**10.31 Restoration**

A degraded, suspended, expired, or revoked Trust-State SHALL NOT automatically return to VALID status merely because the triggering condition no longer exists.

Where restoration is permitted, the Rule-State SHALL define:

- restoration authority;
- required evidence;
- required re-evaluation;
- applicable Rule-State;
- continuity requirements;
- new Trust-State identity, where material;
- integrity requirements; and
- provenance requirements.

**10.32 Reinstatement Following Revocation**

Where reinstatement following revocation is permitted, reinstatement SHALL constitute a governed state transition.

The system SHALL preserve the distinction between:

- prior valid state;
- revoked state; and
- reinstated or newly derived state.

Reinstatement SHALL NOT erase revocation history.

**10.33 Trust-State Identity Across Change**

A material change producing a materially different Trust-State SHALL result in a distinguishable Trust-State identity or version.

The system SHALL preserve relationships between:

- prior state;
- triggering change;
- re-evaluated state;

- resulting classification; and
- applicable provenance.

**10.34 Continuity Evidence**

A conforming system SHALL preserve evidence sufficient to establish continuity decisions.

Continuity evidence SHALL include or reference, where applicable:

- prior Trust-State;
- material change;
- detection event;
- applicable Rule-State;
- re-evaluation procedure;
- resulting Trust-State;
- resulting execution eligibility;
- revocation status;
- continuity classification; and
- integrity information.

**10.35 Continuity Binding**

A continuity determination SHALL be bound to:

- the affected prior Trust-State;
- the material change;
- governing conditions;
- applicable Rule-State;
- resulting determination; and
- effective time.

The binding SHALL prevent material ambiguity regarding why a state remained valid, degraded, suspended, expired, or revoked.

**10.36 Revocation Evidence**

A revocation event SHALL preserve sufficient evidence to establish:

- what was revoked;
- who or what revoked it;
- applicable authority;
- scope;
- effective time;
- governing basis;
- dependent effects;
- enforcement action; and
- relevant integrity information.

**10.37 Independent Verification of Continuity**

Where required by the applicable Conformance Profile, the system SHALL preserve sufficient information for an independent verifier to determine:

- whether material change occurred;
- whether the change was detected;
- whether re-evaluation was required;
- which Rule-State governed;
- whether the prior Trust-State remained valid;
- whether revocation applied;
- whether dependent eligibility was updated; and
- whether enforcement behaved correctly.

**10.38 Independent Recomputation of Continuity**

Where independent recomputation is required, the system SHALL preserve or make reproducibly available the conditions necessary to recreate the continuity determination.

Equivalent:

- prior Trust-State;
- material change;
- Authority-State;
- evidence state;
- Rule-State;
- context;
- dependency state;
- temporal state;
- revocation state; and
- continuity evaluation procedure

SHALL produce an equivalent continuity outcome.

**10.39 Continuity Replay**

Where replay is required, the system SHALL permit reconstruction of the continuity state at a prior point in time.

Replay SHALL preserve the governing conditions and Rule-State applicable to the historical continuity event.

**10.40 Continuity Failure**

A continuity failure occurs where a system continues to rely upon a prior Trust-State after a material change that required invalidation or re-evaluation.

Continuity failures MAY include:

- missed revocation;
- stale authority;
- stale evidence;
- expired Trust-State;
- superseded Rule-State;
- invalid dependency;
- unprocessed material change;

- invalid continued execution; or
- failure to propagate revocation.

**10.41 Continuity Failure Classification**

A conforming implementation MAY classify continuity failures.

Failure classes MAY include:

- STALE_STATE;
- REVOCATION_MISSED;
- REEVALUATION_MISSED;
- EXPIRATION_MISSED;
- PROPAGATION_FAILURE;
- INTEGRITY_FAILURE;
- DEPENDENCY_FAILURE;
- CONTINUED_EXECUTION_AFTER_INVALIDATION; and
- other objectively defined classifications.

**10.42 Retention**

Continuity and revocation records SHALL be preserved for the retention period required by the applicable:

- Rule-State;
- Conformance Profile;
- certification requirement;
- contractual requirement;
- regulatory requirement; or
- implementation policy.

Retention SHALL be sufficient to support required verification, replay, and recomputation.

### 10.43 Conformance Requirement

A system conforms to this section when it can demonstrate that Trust-State validity is:

- bounded;
- monitored for material change where required;
- re-evaluated where required;
- expired where applicable;
- degraded or suspended where defined;
- revoked where governing conditions require;
- propagated to dependent determinations where required;
- integrity-protected;
- preserved with sufficient provenance; and
- independently verifiable or recomputable in accordance with the applicable Conformance Profile.

### 10.44 Canonical Statement

**Trust-State is not permanent authorization.**

Trust-State remains valid only while the conditions that support it remain valid.

When authority, evidence, rules, context, dependencies, or revocation state materially change, continued execution requires continued admissibility.

## Section 11 — Canonical Serialization and Integrity

### 11.0 Purpose

This section defines the normative requirements for canonical representation, deterministic serialization, integrity protection, and representation identity across Trust-State Standard governed objects.

Canonical serialization provides the common deterministic substrate by which governing states, inputs, determinations, and execution evidence can be represented consistently, compared, integrity-protected, independently verified, recomputed, and replayed.

This section is normative.

### 11.1 Canonical Representation Principle

Where deterministic representation is required by the applicable Conformance Profile, a conforming system SHALL represent material governed inputs and outputs under defined canonical rules.

Equivalent governed inputs SHALL produce equivalent canonical representations.

Canonical representation SHALL eliminate implementation-dependent variation that could materially alter:

- representation identity;
- integrity value;
- comparison;
- admissibility evaluation;
- recomputation;
- replay; or
- independent verification.

### 11.2 Objects Subject to Canonical Representation

Canonical representation SHALL apply, where required by the applicable Conformance Profile, to material governed objects including:

- Authority-State;
- evidence;
- evidence sets;
- Rule-State;

- contextual conditions;
- admissibility determinations;
- Trust-State;
- execution eligibility determinations;
- continuity determinations;
- revocation events;
- execution events;
- conformance artifacts; and
- other objects identified by the applicable normative specification.

Not every implementation object is required to be canonicalized.

Canonicalization SHALL apply to the material representation required for deterministic verification.

**11.3 Canonicalization Boundary**

A canonicalization boundary SHALL define which information is included in the canonical representation.

The boundary SHALL be sufficiently explicit to distinguish:

- included fields;
- excluded fields;
- required fields;
- optional fields;
- derived fields;
- metadata;
- external references;
- implementation-specific information; and
- non-governing presentation data.

Material governing information SHALL NOT be excluded from the canonicalization boundary where exclusion would prevent deterministic verification or recomputation.

### 11.4 Deterministic Serialization

Deterministic serialization is the transformation of a governed object into a canonical serialized representation under defined rules.

The serialization procedure SHALL define, where applicable:

- encoding format;
- field ordering;
- object ordering;
- array ordering;
- character encoding;
- whitespace treatment;
- numeric representation;
- boolean representation;
- null representation;
- date and time representation;
- identifier representation;
- binary-data representation;
- escaping rules;
- normalization rules; and
- treatment of optional or absent values.

### 11.5 Equivalent Inputs

Equivalent inputs SHALL produce equivalent serialized outputs under the applicable canonicalization procedure.

For purposes of this section, equivalence means semantic equivalence within the declared canonicalization boundary.

Implementation-specific differences that do not alter governed meaning SHALL NOT produce materially different canonical representations where the applicable serialization specification defines them as equivalent.

**11.6 Canonical Field Identity**

Fields included in a canonical representation SHALL have stable identity.

Field identity SHALL be defined sufficiently to prevent ambiguity arising from:

- alternate names;
- aliases;
- capitalization;
- localization;
- display labels;
- presentation formatting; or
- implementation-specific naming.

Where aliases are permitted, the canonical representation SHALL resolve them to a single canonical field identity.

**11.7 Canonical Field Ordering**

Where field order affects serialization output, the ordering rule SHALL be deterministic.

Canonical ordering MAY be based upon:

- lexicographic ordering;
- schema-defined ordering;
- numeric ordering;
- dependency ordering; or
- another explicitly defined deterministic rule.

The applicable specification SHALL define the ordering method.

**11.8 Canonical Collection Ordering**

Where collections, arrays, sets, lists, or grouped values are included in a canonical representation, the applicable specification SHALL define whether ordering is:

- significant;
- insignificant and therefore normalized;

- schema-defined;
- identifier-based; or
- derived under another deterministic rule.

Semantically equivalent unordered collections SHALL NOT produce different canonical results solely because of implementation-specific ordering.

**11.9 Canonical Numeric Representation**

Where numeric values are included, the serialization procedure SHALL define a deterministic representation.

The procedure SHALL address, where applicable:

- integer form;
- decimal form;
- precision;
- scale;
- sign;
- leading zeros;
- trailing zeros;
- scientific notation;
- rounding;
- overflow;
- underflow; and
- invalid numeric values.

Equivalent numeric values SHALL NOT produce different canonical representations solely because of formatting differences.

**11.10 Canonical Time Representation**

Where time is material, the serialization procedure SHALL define a canonical temporal representation.

The representation SHALL address, where applicable:

- timezone;

- UTC normalization;
- precision;
- offset treatment;
- calendar representation;
- date format;
- timestamp format;
- interval representation; and
- ordering.

A timestamp used to establish governing state SHALL be interpretable independently of local presentation settings.

**11.11 Canonical Identifier Representation**

Identifiers SHALL be represented in a stable and unambiguous form.

Where identifiers may appear in multiple equivalent forms, the canonicalization procedure SHALL define the canonical form.

Identifiers MAY include:

- actor identifiers;
- action identifiers;
- Authority-State identifiers;
- evidence identifiers;
- Rule-State identifiers;
- determination identifiers;
- Trust-State identifiers;
- execution identifiers;
- revocation identifiers; and
- conformance artifact identifiers.

### 11.12 External References

A canonical representation MAY reference external objects rather than embedding them in full.

Where external references are used, the reference SHALL be sufficient to identify the referenced object and verify its integrity where integrity is material.

An external reference MAY include:

- stable identifier;
- version;
- integrity value;
- content address;
- URI;
- ledger reference;
- registry reference; or
- other deterministic pointer.

An external reference SHALL NOT introduce ambiguity regarding which object was relied upon.

### 11.13 Canonical Evidence Sets

Where multiple evidence items are relied upon in a determination, the evidence set SHALL be represented in a manner sufficient to establish exactly which evidence items were used.

The canonical evidence set SHALL preserve or reference:

- evidence identities;
- material versions;
- integrity values where applicable;
- ordering rules where material;
- provenance references where required; and
- sufficiency-relevant attributes.

A change to the material evidence set SHALL produce a distinguishable canonical representation.

### 11.14 Canonical Authority-State Representation

A canonical Authority-State representation SHALL preserve or reference the material authority attributes required by Section 5.

These SHALL include, where applicable:

- source;
- subject;
- scope;
- constraints;
- lineage;
- temporal validity;
- revocation state;
- provenance; and
- Authority-State identity.

A material change to authority conditions SHALL produce a distinguishable representation.

### 11.15 Canonical Rule-State Representation

A canonical Rule-State representation SHALL preserve or reference the material governing attributes required by Section 7.

These SHALL include, where applicable:

- rule identities;
- policy identities;
- versions;
- applicability;
- effective state;
- constraints;
- precedence;
- conflict-resolution rules;
- provenance; and

- Rule-State identity.

A material Rule-State change SHALL produce a distinguishable representation.

### 11.16 Canonical Context Representation

Where context is material to admissibility or execution eligibility, the context SHALL be represented sufficiently to establish the contextual conditions evaluated.

Canonical context MAY include:

- time;
- jurisdiction;
- environment;
- actor state;
- resource state;
- dependency state;
- transaction state;
- system state; and
- other Rule-State-defined context.

Only context material to the governed determination is required to be included.

### 11.17 Canonical Determination Representation

A canonical admissibility or execution eligibility determination SHALL preserve or reference sufficient information to establish:

- determination identity;
- action identity;
- Authority-State identity;
- evidence-set identity;
- Rule-State identity;
- contextual boundary;
- temporal state;

- result;
- applicable Conformance Profile; and
- integrity information.

### 11.18 Trust-State Representation

Where Trust-State is represented canonically, the representation SHALL identify the governed conditions under which the Trust-State was derived.

A Trust-State representation SHALL NOT be detached from the action, scope, Rule-State, and governing inputs material to its derivation.

### 11.19 Continuity and Revocation Representation

Canonical continuity and revocation records SHALL preserve or reference sufficient information to establish:

- affected prior state;
- material change;
- revocation source where applicable;
- effective time;
- applicable Rule-State;
- resulting state;
- dependent effects; and
- integrity information.

### 11.20 Integrity Value

An integrity value is a deterministic value derived from a canonical representation for the purpose of establishing representation identity or detecting material modification.

Where an integrity value is required, it SHALL be generated from the canonical representation specified for the governed object.

The integrity procedure SHALL identify the algorithm and applicable parameters.

### 11.21 Cryptographic Hashing

Where cryptographic hashing is used to derive an integrity value, the applicable specification SHALL define:

- hash algorithm;
- algorithm version where applicable;
- canonical input representation;
- byte encoding;
- domain-separation method where applicable; and
- any required prefixes, namespaces, or type identifiers.

Hashing non-canonical input SHALL NOT satisfy a requirement for canonical integrity verification.

### 11.22 Domain Separation

Where the same hashing mechanism is used across multiple governed object types, the serialization or hashing procedure SHOULD support domain separation.

Domain separation MAY distinguish:

- Authority-State;
- evidence;
- Rule-State;
- admissibility determination;
- Trust-State;
- execution eligibility;
- revocation;
- execution event; and
- conformance artifact.

Domain separation SHALL be deterministic where used.

### 11.23 Integrity Verification

A conforming system SHALL be capable of verifying integrity values required by the applicable Conformance Profile.

Verification SHALL determine whether the governed object corresponds to the canonical representation from which the integrity value was derived.

An integrity mismatch SHALL be treated as an integrity failure.

**11.24 Representation Identity**

A canonical representation MAY serve as a basis for deterministic representation identity.

Representation identity SHALL permit a verifier to determine whether two representations correspond to equivalent governed content under the applicable canonicalization procedure.

Representation identity SHALL NOT imply semantic equivalence outside the declared canonicalization boundary.

**11.25 Material Change and Representation Change**

A material change to a governed object SHALL produce a distinguishable canonical representation.

A non-material change that is outside the canonicalization boundary MAY leave the canonical representation unchanged.

The applicable specification SHALL distinguish material governed content from non-governing presentation or implementation content.

**11.26 Canonicalization and Versioning**

Canonicalization rules SHALL be version-identifiable.

A conforming implementation SHALL identify the canonicalization specification and version used to generate a canonical representation.

A change to the canonicalization procedure that can alter serialized output SHALL result in a distinguishable serialization-specification version.

**11.27 Backward Compatibility**

A new canonicalization version MAY preserve compatibility with a prior version.

Where compatibility is claimed, the applicable specification SHALL define:

- compatible versions;
- compatibility conditions;

- transformation rules;
- prohibited transformations; and
- verification requirements.

Compatibility SHALL NOT be presumed solely because two representations contain similar fields.

**11.28 Cross-Implementation Determinism**

Where cross-implementation reproducibility is required, independent conforming implementations SHALL produce equivalent canonical representations from equivalent governed inputs under the same serialization specification.

Implementation language, platform, library, data store, or operating environment SHALL NOT alter the canonical result.

**11.29 Cross-System Comparison**

Canonical representations MAY be used to compare governed states across systems.

Comparison SHALL be limited to objects represented under compatible canonicalization rules.

A comparison SHALL NOT claim equivalence where:

- canonicalization specifications differ materially;
- governing boundaries differ;
- required fields are absent;
- Rule-State versions differ materially; or
- other conditions required for equivalence are not satisfied.

**11.30 Serialization Failure**

A serialization failure occurs where a governed object cannot be transformed into the required canonical representation.

Serialization failure MAY result from:

- invalid input;
- undefined field treatment;
- unsupported data type;

- ambiguous encoding;
- schema violation;
- invalid timestamp;
- invalid identifier;
- normalization failure; or
- another condition defined by the applicable serialization specification.

Where canonical serialization is required for the determination, serialization failure SHALL prevent assertion of the dependent canonical result.

**11.31 Integrity Failure**

An integrity failure occurs where:

- a required integrity value cannot be generated;
- a required integrity value cannot be verified;
- canonical content does not match its claimed integrity value;
- the applicable algorithm cannot be established;
- serialization version cannot be established; or
- another material integrity condition fails.

Where integrity is required for admissibility, eligibility, continuity, replay, or conformance, an unresolved integrity failure SHALL prevent reliance upon the affected object.

**11.32 Canonicalization Failure Handling**

Failure behavior SHALL be defined by the applicable Rule-State, Conformance Profile, or normative serialization specification.

Where fail-closed behavior is required, inability to establish the required canonical representation or integrity value SHALL prevent the dependent determination from being treated as valid.

**11.33 Preservation of Canonical Representations**

Where required for verification, replay, or recomputation, a conforming system SHALL preserve:

- the canonical representation itself; or

- sufficient governed input and specification information to reproduce it.

Preservation SHALL include the applicable canonicalization version.

**11.34 Preservation of Integrity Values**

Integrity values relied upon in a determination SHALL be preserved or reproducibly derivable for the applicable retention period.

Preserved integrity values SHALL remain associated with the governed object and applicable serialization version.

**11.35 Independent Verification**

Where required by the applicable Conformance Profile, an independent verifier SHALL be able to determine:

- which canonicalization specification applied;
- which version applied;
- the canonicalization boundary;
- which material fields were included;
- the canonical serialized result;
- the integrity algorithm;
- the resulting integrity value; and
- whether the representation verifies correctly.

Independent verification SHALL NOT depend solely upon a system assertion that data was canonicalized correctly.

**11.36 Independent Recomputation**

Where independent recomputation is required, a verifier SHALL be able to regenerate the canonical representation from equivalent governed inputs using the applicable serialization specification.

Equivalent governed inputs SHALL produce an equivalent canonical representation and, where applicable, equivalent integrity value.

### 11.37 Replay Support

Canonical representations SHALL support replay where replay is required by the applicable Conformance Profile.

Replay SHALL use the canonicalization procedure applicable to the historical determination or a verified compatibility procedure.

A current serialization format SHALL NOT silently replace the historical canonicalization procedure where doing so can change the reconstructed result.

### 11.38 Canonical Artifact Binding

Where a machine-verifiable artifact represents a governed determination, the artifact SHOULD bind or reference the canonical identities of the material objects upon which the determination depends.

Such binding MAY include:

- action integrity value;
- Authority-State integrity value;
- evidence-set integrity value;
- Rule-State integrity value;
- context integrity value;
- determination integrity value;
- Trust-State integrity value; and
- execution eligibility integrity value.

The binding structure SHALL be deterministic where required by the applicable Conformance Profile.

### 11.39 Artifact Integrity

A machine-verifiable Trust-State artifact SHALL be integrity-protected where required by the applicable Conformance Profile.

Integrity protection SHALL enable detection of material alteration to the artifact or its governed references.

Artifact integrity SHALL NOT by itself establish that the underlying governing inputs were substantively valid.

**11.40 Integrity and Authenticity Separation**

Integrity and authenticity SHALL be treated as distinct.

Integrity establishes whether represented content has materially changed relative to a protected representation.

Authenticity establishes whether content or an assertion originates from a claimed source.

A valid integrity value SHALL NOT, by itself, establish authenticity.

**11.41 Integrity and Authority Separation**

Integrity SHALL NOT be treated as authority.

A cryptographically intact Authority-State, credential, rule, evidence item, or artifact MAY still lack sufficient authority, applicability, validity, or admissibility.

Integrity protects the governed representation.

It does not determine whether the governed condition is sufficient.

**11.42 Integrity and Evidence Sufficiency Separation**

Evidence integrity SHALL NOT be treated as evidence sufficiency.

An integrity-protected evidence item MAY remain:

- irrelevant;
- expired;
- revoked;
- out of scope;
- incomplete; or
- otherwise insufficient.

The evidence requirements of Section 6 SHALL remain independently applicable.

**11.43 Deterministic Comparison**

Where two governed objects are claimed to be equivalent, the comparison procedure SHALL define:

- the compared canonicalization version;

- the compared boundary;
- identity rules;
- permissible transformations;
- required fields; and
- equivalence criteria.

Hash equality MAY establish representation equality under the applicable procedure but SHALL NOT establish broader semantic or legal equivalence beyond the governed boundary.

**11.44 Conformance Evidence**

A conforming implementation SHALL preserve sufficient evidence to demonstrate compliance with the canonicalization and integrity requirements applicable to its declared Conformance Profile.

Such evidence MAY include:

- canonical test vectors;
- serialization outputs;
- integrity values;
- algorithm identifiers;
- version identifiers;
- implementation test results;
- cross-implementation comparison results;
- failure tests; and
- replay results.

**11.45 Unknown and Out-of-Schema Attributes**

Where a governed object contains an attribute that is unrecognized, unsupported, or outside the applicable object schema, the canonicalization procedure SHALL define its deterministic treatment.

The procedure SHALL determine whether the attribute is:

- rejected;
- excluded as non-governing content;
- preserved as governed content;

- transformed into a defined canonical field; or
- otherwise handled under an explicitly defined rule.

An implementation SHALL NOT silently drop, reinterpret, reorder, or preserve an unknown attribute in a manner capable of causing materially different canonical representations across conforming implementations.

Where the status or treatment of an unknown attribute cannot be determined and the attribute may affect governed meaning, canonical serialization SHALL be treated as failed or indeterminate according to the applicable specification.

**11.46 Conformance Requirement**

A system conforms to this section when it can demonstrate that material governed representations required by the applicable Conformance Profile are:

- canonically bounded;
- deterministically serialized;
- version-identifiable;
- integrity-protected where required;
- reproducible from equivalent inputs;
- comparable under defined rules;
- preservable for replay and recomputation; and
- independently verifiable.

**11.47 Canonical Statement**

**Deterministic governance requires deterministic representation.**

If materially equivalent governing conditions can be represented differently in ways that alter identity, integrity, evaluation, or replay, the determination is not independently reproducible.

Canonical serialization binds governed meaning to a representation that can be verified again.

## Section 12 — Independent Recomputation and Replay

### 12.0 Purpose

This section defines the normative requirements for independently reproducing, verifying, and replaying Trust-State determinations and related governed states.

Independent recomputation and replay provide the mechanism by which a prior determination can be tested without relying upon the original evaluator's assertion.

This section is normative.

### 12.1 Recomputation Principle

Where independent recomputation is required by the applicable Conformance Profile, a conforming system SHALL preserve or make reproducibly available sufficient information for an independent evaluator to recreate the original governed determination.

Recomputation SHALL establish whether equivalent governing conditions and equivalent evaluation logic produce an equivalent result.

### 12.2 Independence

An independent recomputation SHALL be performed by an evaluator that is logically or operationally separate from the original determination process.

Independence MAY be established through:

- separate software implementation;
- separate service;
- separate execution environment;
- separate organizational function;
- third-party evaluator;
- conformance test harness; or
- another method sufficient to prevent blind reliance upon the original result.

Independence SHALL NOT require organizational separation where logical separation is sufficient for the applicable Conformance Profile.

**12.3 Recomputation Scope**

Independent recomputation MAY apply to:

- Authority-State evaluation;
- evidence sufficiency;
- Rule-State selection;
- admissibility determination;
- Trust-State derivation;
- execution eligibility;
- continuity determination;
- revocation effect;
- canonical representation;
- integrity value;
- execution enforcement outcome; and
- other governed determinations identified by the applicable Conformance Profile.

**12.4 Required Recomputation Inputs**

A conforming system SHALL preserve or make reproducibly available the material governing inputs required to recreate the original determination.

These SHALL include, where applicable:

- action identity;
- action request;
- Authority-State;
- evidence set;
- Rule-State;
- context;
- temporal conditions;
- dependency state;
- revocation state;

- applicable constraints;
- evaluation procedure;
- canonicalization specification;
- serialization version; and
- other material inputs.

**12.5 Input Identity**

Inputs used for recomputation SHALL be identifiable.

The system SHALL preserve sufficient information to establish whether the recomputation used:

- the same input;
- an equivalent input;
- a later input;
- a superseding input; or
- a materially different input.

A recomputation claim SHALL NOT represent materially different inputs as equivalent.

**12.6 Historical Governing State**

Historical recomputation SHALL use the governing conditions applicable at the original determination point.

Historical recomputation SHALL preserve or reproduce, where material:

- historical Authority-State;
- historical evidence state;
- historical Rule-State;
- historical context;
- historical dependency state;
- historical revocation state;
- historical canonicalization procedure; and
- historical evaluation procedure.

Current governing state SHALL NOT silently replace historical governing state.

### 12.7 Evaluation Procedure Identity

The evaluation procedure used for recomputation SHALL be identifiable and version-controlled where changes can affect output.

The procedure SHALL define, where applicable:

- input normalization;
- rule selection;
- evaluation order;
- precedence;
- conflict resolution;
- treatment of indeterminate conditions;
- treatment of probabilistic inputs;
- failure behavior; and
- output derivation.

### 12.8 Equivalent Evaluation Procedure

A recomputation MAY use a different implementation from the original evaluator provided that the implementation applies equivalent governing logic.

Equivalent implementation SHALL produce equivalent outputs from equivalent governed inputs.

Implementation language, platform, library, service architecture, or execution environment SHALL NOT alter the governed result.

### 12.9 Recomputation Output

A recomputation SHALL produce an output sufficient to compare against the original determination.

The recomputation output SHALL identify, where applicable:

- recomputed Authority-State condition;
- evidence sufficiency state;
- applicable Rule-State;

- admissibility outcome;
- Trust-State;
- execution eligibility;
- continuity state;
- revocation effect;
- canonical representation;
- integrity value; and
- comparison result.

**12.10 Equivalence**

Recomputation equivalence exists when equivalent governing inputs and equivalent evaluation logic produce an equivalent governed output.

The applicable specification SHALL define the equivalence criteria.

Equivalent outcome SHALL NOT require byte-for-byte equality of non-governing implementation metadata.

**12.11 Determination Equivalence**

Where admissibility or Trust-State is recomputed, the independently derived determination SHALL be equivalent to the original determination under the applicable equivalence criteria.

A mismatch SHALL be detectable.

**12.12 Canonical Representation Equivalence**

Where canonical representation is part of the Conformance Profile, equivalent governed inputs SHALL produce equivalent canonical representations under the applicable serialization specification.

A canonical representation mismatch SHALL be treated as a recomputation discrepancy.

**12.13 Integrity Value Equivalence**

Where an integrity value is deterministically derived from a canonical representation, equivalent canonical representations SHALL produce equivalent integrity values under the same integrity procedure.

An integrity mismatch SHALL trigger verification of:

- governing inputs;
- canonicalization version;
- serialization procedure;
- algorithm;
- parameters; and
- representation integrity.

**12.14 Recomputation Discrepancy**

A recomputation discrepancy occurs where the independently derived result does not satisfy the applicable equivalence criteria.

A discrepancy MAY result from:

- input mismatch;
- Rule-State mismatch;
- evaluation-procedure mismatch;
- serialization mismatch;
- implementation defect;
- nondeterministic behavior;
- integrity failure;
- incomplete preserved state;
- undisclosed dependency; or
- another material difference.

**12.15 Discrepancy Classification**

A conforming implementation MAY classify recomputation discrepancies.

Classes MAY include:

- INPUT_MISMATCH;
- AUTHORITY_STATE_MISMATCH;

- EVIDENCE_STATE_MISMATCH;
- RULE_STATE_MISMATCH;
- CONTEXT_MISMATCH;
- PROCEDURE_MISMATCH;
- SERIALIZATION_MISMATCH;
- INTEGRITY_MISMATCH;
- OUTPUT_MISMATCH;
- NONDETERMINISTIC_RESULT; and
- INSUFFICIENT_RECOMPUTATION_DATA.

Classification criteria SHALL be objectively determinable where used.

**12.16 Discrepancy Handling**

Where a material recomputation discrepancy occurs, the system SHALL treat the affected determination according to the applicable Rule-State or Conformance Profile.

The system MAY require:

- investigation;
- suspension;
- invalidation;
- re-evaluation;
- corrected artifact issuance;
- conformance failure;
- escalation; or
- another governed response.

A material discrepancy SHALL NOT be silently ignored.

**12.17 Replay Principle**

Replay is the reconstruction and re-evaluation of a prior governed determination using the governing conditions applicable to that prior determination.

Replay SHALL permit verification of what the system should have determined under the historical conditions presented.

**12.18 Historical Replay**

Historical replay SHALL reproduce or make reproducibly available the conditions applicable at the original evaluation point.

Historical replay SHALL preserve:

- original action;
- historical Authority-State;
- historical evidence;
- historical Rule-State;
- historical context;
- historical temporal state;
- historical dependencies;
- historical revocation state; and
- applicable historical evaluation procedure.

**12.19 Replay-Equivalent Evaluation**

Replay-equivalent evaluation exists where equivalent preserved governing conditions and equivalent evaluation logic produce an equivalent historical result.

Replay equivalence SHALL be determined according to defined criteria.

**12.20 Replay and Current Re-Evaluation Separation**

Replay and current re-evaluation SHALL be treated as distinct processes.

Replay asks:

**What should the system have determined under the governing conditions applicable then?**

Current re-evaluation asks:

**What should the system determine under the governing conditions applicable now?**

The two processes MAY produce different valid results.

**12.21 Replay of Superseded Rule-State**

Where the historical Rule-State has been superseded, replay SHALL use the historical Rule-State or a verified equivalent representation.

A successor Rule-State SHALL NOT be substituted merely because it is currently active.

**12.22 Replay of Revoked Inputs**

Historical replay MAY require use of an input that is currently revoked.

Where so, the replay SHALL preserve the historical revocation state applicable at the original evaluation time.

Current revocation SHALL NOT retroactively alter the historical replay result unless the governing rules explicitly require retrospective effect.

**12.23 Replay of Expired Evidence**

Evidence currently expired MAY remain valid for historical replay if it was valid at the original evaluation point.

Replay SHALL distinguish historical validity from current validity.

**12.24 Replay of Changed Authority**

Authority currently modified, expired, or revoked MAY be used in historical replay where the historical Authority-State remains reproducibly identifiable.

Replay SHALL evaluate authority as it existed at the relevant historical point.

**12.25 Replay and Material Change**

A material change occurring after the original determination SHALL NOT alter historical replay unless the Rule-State assigns retrospective effect to that change.

Such change MAY, however, affect current continuity or current admissibility.

**12.26 Replay Data Preservation**

Where replay is required, a conforming system SHALL preserve or reproducibly reconstruct sufficient historical state to perform replay.

Replay data SHALL include or reference:

- action;
- Authority-State;
- evidence;
- Rule-State;
- context;
- temporal state;
- dependency state;
- revocation state;
- evaluation procedure;
- canonicalization specification;
- integrity values; and
- original result.

**12.27 Replay Artifact**

A conforming implementation MAY generate a replay artifact.

A replay artifact SHOULD include or reference:

- original determination identity;
- replay identity;
- historical governing inputs;
- applicable Rule-State;
- replay procedure;
- recomputed result;
- equivalence result;
- discrepancy classification where applicable; and
- integrity information.

### 12.28 Replay Integrity

Replay data SHALL be integrity-protected where required by the applicable Conformance Profile.

A replay result SHALL NOT be treated as reliable where the historical governing state necessary for replay cannot be shown to be authentic or integrity-preserved to the required level.

### 12.29 Independent Verification

An independent verifier SHALL be able to determine, where required:

- what original action was evaluated;
- what governing inputs were used;
- which Rule-State applied;
- which procedure applied;
- which canonicalization specification applied;
- what original result was produced;
- what recomputed result was produced;
- whether the results are equivalent; and
- whether a discrepancy exists.

### 12.30 Independent Challenge

A conforming system MAY support independent challenge of a prior determination.

A challenge MAY assert:

- incorrect governing input;
- incorrect Rule-State;
- incorrect authority evaluation;
- incorrect evidence evaluation;
- incorrect context;
- incorrect procedure;
- serialization error;
- integrity failure;

- deterministic mismatch; or
- another material defect.

Challenge handling SHALL itself be governed where implemented.

**12.31 Verification Without Trust in Original Evaluator**

Independent recomputation SHALL be designed so that verification does not require trust in the original evaluator's conclusion.

The original determination MAY be used as a comparison target.

It SHALL NOT be treated as proof that the determination was correct.

**12.32 Verification Without Original Execution Environment**

Where required by the applicable Conformance Profile, recomputation SHOULD be possible without reproducing the entire original execution environment.

A conforming system SHOULD preserve the governed state necessary for deterministic evaluation rather than relying upon undocumented environmental state.

Where environmental conditions are material, they SHALL be represented as governed inputs.

**12.33 Hidden State Prohibition**

A deterministic recomputation SHALL NOT depend upon material hidden state that is unavailable to the independent evaluator.

Material hidden state includes any unrepresented condition that can alter the governed outcome.

Where such state is necessary, it SHALL be included or referenced within the recomputation boundary.

**12.34 Undisclosed External Dependency**

A conforming system SHALL identify external dependencies capable of materially altering a determination.

Independent recomputation SHALL NOT depend upon an undisclosed external source whose historical state cannot be reproduced or verified.

**12.35 Nondeterministic Components**

A system MAY contain nondeterministic or probabilistic components.

Where such components contribute to a governed determination, the system SHALL preserve the material outputs of those components or otherwise bound them such that the governing determination remains independently reproducible under the applicable Conformance Profile.

Nondeterminism outside the governed determination boundary does not by itself prevent conformity.

**12.36 Probabilistic Inputs in Replay**

Where probabilistic outputs were used as governed inputs in the original determination, historical replay SHALL use the preserved historical output or another representation defined by the Rule-State as equivalent.

Replay SHALL NOT require a probabilistic model to regenerate the identical historical score unless the applicable specification expressly requires such regeneration.

**12.37 Human Inputs in Replay**

Where a human decision formed part of the historical governed input, replay SHALL preserve or reference the governed human decision and its applicable provenance.

Replay SHALL NOT require an independent evaluator to reproduce the human's internal reasoning unless such reasoning was itself a required governed input.

**12.38 Execution Replay**

Where required, a conforming system SHALL permit reconstruction of whether execution should have been permitted at the historical execution boundary.

Execution replay SHALL evaluate:

- historical admissibility;
- historical Trust-State;
- historical execution eligibility;
- historical material changes;
- historical revocation state; and
- applicable enforcement logic.

**12.39 Enforcement Verification**

Independent replay SHALL permit determination of whether the enforcement mechanism:

- correctly permitted eligible execution;
- correctly prevented non-eligible execution;
- failed closed where required;
- responded to revocation;
- responded to material change; and
- applied the correct execution boundary.

**12.40 Continuity Replay**

Where continuity governance is part of the applicable Conformance Profile, replay SHALL permit reconstruction of continuity determinations.

The evaluator SHALL be able to determine:

- prior Trust-State;
- material change;
- applicable continuity Rule-State;
- required response;
- actual response; and
- resulting state.

**12.41 Revocation Replay**

Where revocation materially affected a prior state, replay SHALL permit determination of:

- revocation source;
- effective time;
- affected scope;
- propagation requirement;
- resulting Trust-State;
- resulting eligibility; and

- resulting enforcement behavior.

**12.42 Cross-Implementation Replay**

Where cross-implementation reproducibility is required, two conforming independent evaluators using equivalent governed inputs and equivalent evaluation logic SHALL produce equivalent replay results.

A failure to do so SHALL constitute a deterministic equivalence discrepancy.

**12.43 Retention Boundary**

A conforming system SHALL define the period during which recomputation and replay are supported where such capabilities are required.

The retention boundary MAY be governed by:

- Rule-State;
- Conformance Profile;
- certification requirements;
- legal obligations;
- regulatory requirements;
- contractual requirements; or
- implementation policy.

A conformance claim SHALL NOT imply replay capability beyond the declared retention boundary.

**12.44 Incomplete Historical State**

Where the historical state necessary for replay is incomplete, the system SHALL identify the replay as incomplete, indeterminate, or otherwise non-equivalent according to the applicable specification.

A system SHALL NOT claim successful deterministic replay where required material governing state is unavailable.

**12.45 Replay Failure**

Replay failure occurs where the historical determination cannot be independently reconstructed to the degree required by the applicable Conformance Profile.

Replay failure MAY result from:

- missing historical state;
- missing Rule-State;
- missing evidence;
- missing Authority-State;
- missing context;
- missing evaluation procedure;
- serialization incompatibility;
- integrity failure;
- hidden dependency; or
- nondeterministic evaluation behavior.

**12.46 Recomputation Evidence**

A conforming system SHALL preserve sufficient evidence to demonstrate independent recomputation capability.

Such evidence MAY include:

- recomputation test vectors;
- replay test vectors;
- independent implementation results;
- canonical representations;
- integrity values;
- historical reconstruction results;
- discrepancy reports; and
- equivalence reports.

**12.47 Conformance Requirement**

A system conforms to this section when it can demonstrate, in accordance with the applicable Conformance Profile, that governed determinations can be:

- independently reconstructed;
- evaluated using identifiable governing inputs;
- evaluated under the applicable historical Rule-State;
- reproduced using equivalent evaluation logic;
- compared under defined equivalence criteria;
- replayed without silent substitution of current conditions;
- checked for deterministic discrepancies; and
- verified without relying solely upon the original evaluator's assertion.

**12.48 Canonical Statement**

**A determination is not independently verifiable merely because its result was recorded.**

It is independently verifiable when another evaluator can reconstruct the governing state, apply the applicable rules, and determine whether the same result follows.

Replay proves what should have been determined then. Recomputation proves that the determination can be derived again.

## Section 13 — Conformance Profiles

### 13.0 Purpose

This section defines the Trust-State Standard Conformance Profiles used to declare and evaluate implementation capability.

Conformance Profiles establish distinct levels of governed capability without requiring every conforming implementation to perform execution enforcement, continuity governance, or independent recomputation.

This section is normative.

### 13.1 Conformance Profile Principle

A conformance claim SHALL identify the Conformance Profile against which conformity is asserted.

A higher Conformance Profile SHALL include the requirements of lower profiles where those requirements are expressly incorporated.

Conformance SHALL be determined against:

- the applicable Trust-State Standard release;
- the declared Conformance Profile;
- the declared conformance boundary; and
- the normative requirements applicable to that profile.

A conforming implementation SHALL NOT imply capability beyond its declared Conformance Profile.

### 13.2 Profile Structure

The Trust-State Standard defines three core Conformance Profiles:

- **Profile I — Deterministic Admissibility**
- **Profile II — Deterministic Enforcement**
- **Profile III — Deterministic Continuity and Recomputation**

These profiles represent increasing capability at the consequence-bearing execution boundary.

### 13.3 Profile I — Deterministic Admissibility

Profile I applies to systems that deterministically evaluate whether a proposed consequence-bearing action satisfies applicable governing conditions.

A Profile I implementation SHALL:

- identify the proposed action;
- identify applicable Authority-State;
- determine authority sufficiency where required;
- identify required evidence;
- determine evidence sufficiency where required;
- identify the applicable Rule-State;
- evaluate applicable context;
- evaluate applicable constraints;
- evaluate applicable revocation state;
- derive an admissibility determination;
- derive a Trust-State where required;
- preserve the material governing inputs relied upon in the determination;
- preserve sufficient determination provenance; and
- support independent verification of the determination to the extent required by this profile.

### 13.4 Profile I Applicable Sections

Profile I SHALL require conformity with the applicable requirements of:

- Section 1 — Scope;
- Section 2 — Normative References;
- Section 3 — Terms and Definitions;
- Section 4 — Consequence and Admissibility Model;
- Section 5 — Authority-State Requirements;
- Section 6 — Evidence Requirements;
- Section 7 — Rule-State and Policy Identity;

- Section 8 — Admissibility Determination;
- Section 11 — Canonical Serialization and Integrity, where required by the applicable conformance specification; and
- applicable conformance, governance, and security requirements established elsewhere in the Standard.

**13.5 Profile I Exclusions**

Profile I does not by itself require:

- execution gating;
- pre-execution enforcement;
- execution-event binding;
- continuity monitoring;
- automatic revocation propagation;
- independent recomputation;
- deterministic replay; or
- cross-implementation replay equivalence.

A Profile I determination SHALL NOT be represented as evidence that the system prevented non-admissible execution.

**13.6 Profile I Conformance Statement**

A conforming Profile I implementation demonstrates that it can determine, in an objectively verifiable manner, whether the governing conditions applicable to a proposed consequence-bearing action are satisfied.

Profile I establishes deterministic admissibility.

It does not establish deterministic enforcement.

**13.7 Profile II — Deterministic Enforcement**

Profile II applies to systems that satisfy Profile I and additionally enforce execution eligibility before consequence-bearing execution is permitted to occur.

A Profile II implementation SHALL satisfy all Profile I requirements.

A Profile II implementation SHALL additionally:

- derive execution eligibility from admissibility;
- define the applicable execution boundary;
- enforce execution eligibility before consequence-bearing execution;
- prevent execution where eligibility is absent;
- fail closed where required governing conditions cannot be established;
- prevent execution under expired, revoked, or otherwise invalid eligibility;
- enforce action and scope binding;
- detect execution without valid eligibility;
- preserve enforcement evidence;
- bind execution events to the eligibility determination where required; and
- support independent verification of enforcement behavior.

**13.8 Profile II Applicable Sections**

Profile II SHALL require conformity with the applicable requirements of:

- all Profile I sections; and
- Section 9 — Execution Eligibility and Enforcement.

Where execution enforcement depends upon canonical binding or integrity values, the applicable requirements of Section 11 SHALL apply.

**13.9 Profile II Failure Behavior**

A Profile II implementation SHALL fail closed where the applicable Rule-State or Conformance Profile requires established admissibility and execution eligibility before execution.

Failure to establish:

- required authority;
- required evidence;
- applicable Rule-State;
- required context;

- required integrity;
- required revocation state; or
- another mandatory governing condition

SHALL prevent execution where the condition is required for eligibility.

**13.10 Profile II Enforcement Requirement**

A Profile II implementation SHALL demonstrate that the execution boundary cannot be crossed through the governed pathway without a valid eligibility determination, except where an explicitly governed override, exception, or emergency procedure applies.

Ungoverned bypass capability SHALL constitute non-conformity where it permits consequence-bearing execution outside the required enforcement path.

**13.11 Profile II Conformance Statement**

A conforming Profile II implementation demonstrates that it can both:

- determine whether an action is admissible; and
- prevent the action from becoming consequence where valid execution eligibility has not been established.

Profile II establishes deterministic enforcement.

**13.12 Profile III — Deterministic Continuity and Recomputation**

Profile III applies to systems that satisfy Profile II and additionally govern Trust-State validity across material change and support independent recomputation and replay.

A Profile III implementation SHALL satisfy all Profile I and Profile II requirements.

A Profile III implementation SHALL additionally:

- detect material change where required;
- re-evaluate Trust-State validity when required;
- detect expiration;
- detect applicable revocation;
- propagate revocation where required;

- invalidate or degrade dependent eligibility where required;
- prevent continued execution under stale governing conditions;
- preserve continuity provenance;
- preserve historical governing state required for replay;
- support independent recomputation of applicable determinations;
- support historical replay;
- distinguish replay from current re-evaluation;
- detect deterministic discrepancies; and
- support independent verification of continuity and replay behavior.

**13.13 Profile III Applicable Sections**

Profile III SHALL require conformity with the applicable requirements of:

- all Profile I sections;
- all Profile II sections;
- Section 10 — Trust-State Continuity and Revocation;
- Section 11 — Canonical Serialization and Integrity; and
- Section 12 — Independent Recomputation and Replay.

**13.14 Profile III Continuity Requirement**

A Profile III implementation SHALL NOT presume continued Trust-State validity following material change.

Where a material change can alter admissibility or execution eligibility, the implementation SHALL:

- detect the change;
- apply the applicable continuity rule;
- re-evaluate where required;
- update dependent Trust-State;
- update execution eligibility; and
- enforce the resulting state.

### 13.15 Profile III Recomputation Requirement

A Profile III implementation SHALL preserve or make reproducibly available the material governing state necessary to independently recompute the determinations within its declared conformance boundary.

Equivalent governing inputs and equivalent evaluation logic SHALL produce equivalent governed outputs under the applicable deterministic procedure.

### 13.16 Profile III Replay Requirement

A Profile III implementation SHALL support replay of historical determinations within the declared replay and retention boundary.

Replay SHALL preserve or reproduce:

- historical Authority-State;
- historical evidence;
- historical Rule-State;
- historical context;
- historical revocation state;
- historical dependency state;
- applicable historical canonicalization; and
- applicable historical evaluation procedure.

### 13.17 Profile III Conformance Statement

A conforming Profile III implementation demonstrates that it can:

- determine admissibility;
- enforce execution eligibility;
- maintain or re-evaluate Trust-State across material change; and
- independently reproduce prior governed determinations.

Profile III establishes deterministic continuity and recomputation.

### 13.18 Profile Inheritance

Profile II SHALL inherit all mandatory requirements of Profile I unless a requirement is expressly identified as inapplicable.

Profile III SHALL inherit all mandatory requirements of Profile I and Profile II unless a requirement is expressly identified as inapplicable.

A higher-profile implementation SHALL NOT be required to make separate lower-profile conformance claims where inheritance has been demonstrated.

### 13.19 Partial Implementation

Partial implementation of a Conformance Profile SHALL NOT constitute conformity with that profile.

An implementation MAY declare conformity with a lower profile where:

- all requirements of that lower profile are satisfied; and
- the conformance claim accurately identifies the lower profile.

An implementation SHALL NOT claim Profile II or Profile III conformity merely because selected controls from those profiles are present.

### 13.20 Profile-Specific Conformance Boundary

Each Conformance Profile claim SHALL define its conformance boundary.

The boundary SHALL identify, where applicable:

- implementation or system;
- action class;
- execution pathway;
- actor class;
- Authority-State boundary;
- evidence boundary;
- Rule-State boundary;
- contextual boundary;
- temporal boundary;
- execution boundary;

- continuity boundary;
- replay boundary; and
- retention boundary.

**13.21 Action-Class Conformance**

A system MAY conform for one action class without conforming for all actions supported by the system.

A conformance claim SHALL identify the action classes within scope.

Conformance for one action class SHALL NOT imply conformity for another action class where governing conditions or execution controls materially differ.

**13.22 Component Conformance**

A component MAY claim conformity where the applicable Conformance Profile can be fully satisfied within the component's declared boundary.

Where conformity depends upon external components, the claim SHALL identify the required external dependencies.

A component SHALL NOT claim full-profile conformity where material profile requirements are performed outside the declared boundary and cannot be independently verified.

**13.23 Distributed Conformance**

A Conformance Profile MAY be satisfied across multiple coordinated components.

Where requirements are distributed, the conformance claim SHALL identify:

- component responsibilities;
- governed interfaces;
- evidence exchange;
- Authority-State exchange;
- Rule-State exchange;
- enforcement responsibility;
- continuity responsibility;
- recomputation responsibility; and

- integrity boundaries.

The combined implementation SHALL satisfy all requirements applicable to the claimed profile.

**13.24 Human-in-the-Loop Conformance**

A system MAY use human review, approval, judgment, or intervention within any Conformance Profile where expressly governed.

Human participation SHALL NOT remove requirements for:

- authority;
- evidence;
- Rule-State;
- provenance;
- action binding;
- applicable enforcement; or
- independent verification.

Where human judgment affects deterministic evaluation, the governing boundaries of that judgment SHALL be declared.

**13.25 Autonomous-System Conformance**

Autonomous agents and machine actors MAY conform to any profile.

Autonomy SHALL NOT alter the applicable requirements for:

- Authority-State;
- evidence;
- Rule-State;
- admissibility;
- execution eligibility;
- continuity; or
- recomputation.

A system SHALL NOT claim a reduced governance requirement solely because execution is autonomous.

### 13.26 Probabilistic-System Conformance

A system containing probabilistic or nondeterministic components MAY conform where the governed determination satisfies the deterministic requirements of the applicable profile.

Probabilistic components MAY provide governed inputs.

The conformance claim SHALL identify how those inputs are:

- bounded;
- preserved where required;
- interpreted;
- incorporated into deterministic evaluation; and
- replayed where required.

### 13.27 Profile Declaration

A conformance declaration SHALL identify at minimum:

- Trust-State Standard version;
- Conformance Profile;
- conformance boundary;
- implementation identifier;
- applicable normative annexes;
- canonicalization specification where applicable;
- evaluation-procedure version;
- declared exclusions where permitted; and
- assessment date or assessment state.

### 13.28 Conformance Evidence

Each profile SHALL require evidence sufficient to demonstrate satisfaction of its applicable requirements.

Profile I evidence SHALL include evidence of deterministic admissibility evaluation.

Profile II evidence SHALL additionally include evidence of execution enforcement.

Profile III evidence SHALL additionally include evidence of continuity governance, independent recomputation, and replay.

### 13.29 Profile I Evidence

Profile I conformance evidence SHALL include or reference, where applicable:

- action identification;
- Authority-State;
- authority-sufficiency result;
- evidence set;
- evidence-sufficiency result;
- Rule-State identity;
- context;
- admissibility determination;
- Trust-State;
- determination provenance; and
- integrity evidence required by the declared profile implementation.

### 13.30 Profile II Evidence

Profile II conformance evidence SHALL include all Profile I evidence and additionally:

- execution eligibility;
- execution boundary;
- enforcement decision;
- enforcement mechanism evidence;
- failure-handling evidence;
- execution-event binding where required;
- attempted bypass tests; and
- evidence that non-eligible actions are prevented.

### 13.31 Profile III Evidence

Profile III conformance evidence SHALL include all Profile I and Profile II evidence and additionally:

- material-change events;
- continuity determinations;
- revocation events;
- propagation evidence;
- re-evaluation results;
- historical governing state;
- recomputation results;
- replay results;
- equivalence results; and
- discrepancy-handling evidence.

### 13.32 Negative Testing

Conformance testing SHALL include negative conditions sufficient to demonstrate that the implementation does not merely succeed under valid inputs.

Testing SHOULD include, where applicable:

- missing authority;
- invalid authority;
- revoked authority;
- insufficient evidence;
- revoked evidence;
- invalid Rule-State;
- conflicting rules;
- expired conditions;
- invalid context;
- failed dependencies;
- serialization failure;

- integrity failure;
- eligibility expiration;
- material change;
- missed revocation;
- replay discrepancy; and
- attempted enforcement bypass.

**13.33 Deterministic Testability**

Requirements used to establish profile conformity SHALL be testable through objectively determinable evidence.

A conformance result SHALL NOT depend solely upon:

- vendor assertion;
- policy statement;
- architectural intention;
- design documentation;
- reputation;
- subjective reviewer opinion; or
- claimed best practice.

**13.34 Conformance and Certification Separation**

Conformance and certification SHALL remain distinct.

Conformance is satisfaction of the normative requirements applicable to a declared profile and boundary.

Certification is a formal third-party determination regarding that conformity.

An implementation MAY conform without being certified unless certification is required by another governing requirement.

### 13.35 Certification Profile Identification

Where certification is issued, the certification designation SHALL identify the applicable Conformance Profile.

A certification claim SHALL NOT use a generalized “Trust-State Certified” designation without identifying the governed profile and boundary in the associated certification record.

### 13.36 Certification Inheritance

Certification under a higher Conformance Profile MAY satisfy lower-profile certification requirements where:

- the higher profile formally incorporates the lower requirements;
- the same conformance boundary applies; and
- the certification program permits inheritance.

Inheritance SHALL be documented.

### 13.37 Conformance Validity

A conformance assessment SHALL apply only to the implementation state, profile, and boundary assessed.

Material changes MAY require:

- impact assessment;
- partial reassessment;
- full reassessment; or
- certification renewal

as defined by the applicable governance or certification program.

### 13.38 Profile Change

An implementation MAY move between Conformance Profiles.

A move to a higher profile SHALL require demonstration of all additional applicable requirements.

A move to a lower profile SHALL be reflected in subsequent conformance claims where higher-profile requirements are no longer satisfied.

**13.39 Non-Conformity**

Non-conformity exists where an implementation:

- fails a mandatory requirement;
- claims an unsupported profile;
- exceeds its declared conformance boundary;
- cannot produce required conformance evidence;
- relies upon ungoverned discretionary behavior for a mandatory deterministic condition;
- cannot demonstrate required enforcement;
- cannot demonstrate required continuity;
- cannot demonstrate required recomputation; or
- otherwise fails the applicable profile requirements.

**13.40 Conformance Status**

A conformity assessment MAY classify status using objectively defined categories including:

- CONFORMING;
- CONDITIONALLY_CONFORMING;
- NON_CONFORMING;
- ASSESSMENT_INCOMPLETE;
- SUSPENDED; or
- other certification-program-defined states.

Where conditional conformity is permitted, the applicable program SHALL define the conditions under which such status may be used.

**13.41 No Implication of Regulatory Compliance**

Conformance with a Trust-State Standard Conformance Profile SHALL NOT, by itself, establish:

- legal compliance;
- regulatory approval;
- statutory authorization;

- safety certification;
- cybersecurity certification;
- privacy compliance; or
- another external compliance status.

External requirements MAY reference or rely upon TSS conformance separately.

**13.42 No Implication of Outcome Correctness**

Conformance SHALL NOT imply that every substantive policy decision, authority grant, evidence source, or governing rule is correct, ethical, lawful, or desirable.

Conformance establishes that the declared governing conditions are evaluated, enforced, maintained, and reproduced according to the requirements of the applicable profile.

**13.43 Conformance Profile Summary**

**Profile I — Deterministic Admissibility**

Determines whether a proposed consequence-bearing action satisfies applicable governing conditions.

**Profile II — Deterministic Enforcement**

Determines admissibility and prevents non-eligible actions from crossing the execution boundary.

**Profile III — Deterministic Continuity and Recomputation**

Determines admissibility, enforces execution, governs continued validity across material change, and supports independent recomputation and replay.

**13.44 Normative Annex Applicability**

Normative annex requirements apply to a Conformance Profile only where the annex or an applicable normative requirement expressly identifies that Profile or conformance boundary.

Annex A SHALL apply where canonical serialization is required.

Annex B SHALL apply according to the Profile applicability rules defined in Annex B.

Annex C SHALL apply to governed evaluation procedures within the declared Conformance Profile.

Annex D SHALL apply where machine-readable artifacts or interoperability claims are within the declared conformance boundary.

Annex E applies only to Trust-State certification programs and does not independently expand implementation conformance requirements.

Annex F is informative.

**13.45 Cross-Profile Governance-State Reliance**

A conforming implementation MAY consume a governance state, determination, artifact, or other governed output produced by another implementation operating under a different Conformance Profile.

A higher-profile implementation SHALL NOT infer satisfaction of its own applicable requirements solely from the Conformance Profile or conformance status of the supplying implementation.

Before relying upon an externally supplied governed output, the consuming implementation SHALL establish, to the extent required by its declared Conformance Profile and conformance boundary, that the material input:

- is identifiable;
- applies to the intended action and scope;
- satisfies required integrity conditions;
- was produced under an applicable Rule-State;
- satisfies required authority and evidence conditions where material;
- remains temporally valid;
- is not invalidated by applicable revocation or material change; and
- is sufficient for the consuming implementation's own governed determination.

A lower-profile output MAY satisfy an input requirement of a higher-profile implementation where the required invariants are independently established.

**13.46 Canonical Statement**

**Conformance is not a maturity score.**

It is a declaration of demonstrated capability within a defined boundary.

Profile I determines.

Profile II determines and enforces.

Profile III determines, enforces, maintains, and reproduces.

## Section 14 — Conformance Claims and Evidence

### 14.0 Purpose

This section defines the normative requirements for making, supporting, preserving, and independently verifying claims of conformity with the Trust-State Standard.

A conformance claim SHALL identify the capability, boundary, evidence, and Trust-State Standard version upon which the claim depends.

This section is normative.

### 14.1 Conformance Claim Principle

A conformance claim SHALL be specific, bounded, and evidence-supported.

A claim SHALL NOT imply that an implementation conforms:

- beyond its declared Conformance Profile;
- beyond its declared conformance boundary;
- beyond the assessed implementation state;
- beyond the applicable Trust-State Standard release; or
- beyond the evidence supporting the claim.

Generalized or unbounded claims of Trust-State conformity SHALL NOT satisfy this section.

### 14.2 Required Claim Elements

A conformance claim SHALL identify at minimum:

- Trust-State Standard version;
- Conformance Profile;
- implementation or system identifier;
- conformance boundary;
- assessed action class or action classes;
- applicable Rule-State boundary;
- applicable Authority-State boundary;
- applicable evidence boundary;

- applicable contextual boundary;
- assessment date or assessment state;
- applicable normative annexes;
- canonicalization specification where required; and
- claim status.

**14.3 Standard Version Identification**

A conformance claim SHALL identify the specific Trust-State Standard release against which conformity was assessed.

A claim SHALL NOT state conformity merely to “the Trust-State Standard” where the applicable release is not otherwise identifiable.

Where a claim remains valid after publication of a later Standard release, the original assessed release SHALL remain identified.

**14.4 Conformance Profile Identification**

A conformance claim SHALL identify the applicable Conformance Profile.

The claim SHALL distinguish between:

- Profile I — Deterministic Admissibility;
- Profile II — Deterministic Enforcement; and
- Profile III — Deterministic Continuity and Recomputation.

A claim SHALL NOT imply higher-profile capability unless all applicable requirements of that higher profile have been demonstrated.

**14.5 Conformance Boundary Identification**

A conformance claim SHALL identify the boundary within which the claim applies.

The boundary SHALL identify, where applicable:

- implementation;
- component;
- service;

- action class;
- workflow;
- actor class;
- resource class;
- Authority-State scope;
- evidence scope;
- Rule-State scope;
- contextual scope;
- jurisdictional scope;
- temporal scope;
- execution boundary;
- continuity boundary;
- replay boundary; and
- retention boundary.

**14.6 Action-Class Claim Scope**

Where an implementation supports multiple action classes, a conformance claim SHALL identify which action classes are within scope.

Conformance for one action class SHALL NOT imply conformity for another where:

- governing conditions differ;
- authority requirements differ;
- evidence requirements differ;
- Rule-State differs;
- execution enforcement differs; or
- continuity or recomputation capability differs.

**14.7 Component Claims**

A component MAY make a conformance claim where all applicable requirements of the claimed profile are satisfied within the declared component boundary.

Where conformity depends upon external components, the claim SHALL identify those dependencies.

A component SHALL NOT represent itself as independently conforming to requirements materially performed by an external dependency unless the dependency relationship is included within the declared conformance boundary.

**14.8 Distributed-System Claims**

A distributed implementation MAY make a conformance claim across multiple components.

The claim SHALL identify:

- component identities;
- component responsibilities;
- governed interfaces;
- Authority-State exchange;
- evidence exchange;
- Rule-State exchange;
- execution enforcement responsibility;
- continuity responsibility;
- recomputation responsibility; and
- integrity boundaries.

The combined system SHALL satisfy the applicable profile requirements within the declared boundary.

**14.9 Claim Status**

A conformance claim SHALL identify its status.

Status MAY include:

- CONFORMING;
- CONDITIONALLY_CONFORMING;
- NON_CONFORMING;

- ASSESSMENT_INCOMPLETE;
- SUSPENDED;
- WITHDRAWN; or
- another status defined by the applicable conformance or certification program.

Status criteria SHALL be objectively defined.

**14.10 Conditional Conformance**

Conditional conformity MAY be used only where permitted by the applicable conformance or certification program.

A conditional claim SHALL identify:

- unsatisfied or provisional condition;
- permitted scope of the claim;
- remediation requirement;
- expiration date or review condition;
- restrictions;
- evidence supporting the conditional determination; and
- condition causing the claim to terminate.

Conditional conformity SHALL NOT be used to conceal a material failure of a mandatory requirement.

**14.11 Claim Evidence Principle**

A conformance claim SHALL be supported by evidence sufficient to demonstrate satisfaction of the applicable normative requirements.

Evidence SHALL be:

- attributable;
- integrity-protected where required;
- traceable to the assessed implementation;
- traceable to the applicable Standard version;
- traceable to the applicable Conformance Profile; and

- sufficient for independent verification.

**14.12 Evidence Mapping**

Conformance evidence SHALL be mapped to applicable normative requirements.

The mapping SHALL identify, where applicable:

- requirement identifier;
- evidence item;
- test procedure;
- test result;
- assessment result;
- exception or limitation; and
- reviewer or verification record.

A conformance claim SHALL NOT depend solely upon an undifferentiated evidence archive.

**14.13 Profile I Evidence**

A Profile I claim SHALL be supported by evidence demonstrating, where applicable:

- action identification;
- Authority-State representation;
- authority sufficiency;
- evidence requirements;
- evidence sufficiency;
- Rule-State identity;
- Rule-State applicability;
- contextual evaluation;
- admissibility determination;
- Trust-State derivation;
- determination provenance; and
- required integrity verification.

**14.14 Profile II Evidence**

A Profile II claim SHALL include all required Profile I evidence and additionally demonstrate:

- execution eligibility derivation;
- execution boundary definition;
- pre-execution enforcement;
- fail-closed behavior where required;
- prevention of non-eligible execution;
- action-to-eligibility binding;
- enforcement-event evidence;
- execution-event binding where required;
- revocation-before-execution handling; and
- attempted bypass testing.

**14.15 Profile III Evidence**

A Profile III claim SHALL include all required Profile I and Profile II evidence and additionally demonstrate:

- material-change detection;
- continuity evaluation;
- Trust-State re-evaluation;
- expiration handling;
- revocation handling;
- revocation propagation where required;
- stale-state prevention;
- historical governing-state preservation;
- independent recomputation;
- historical replay;
- deterministic equivalence;
- discrepancy detection; and

- discrepancy handling.

**14.16 Positive Testing**

Conformance testing SHALL include positive conditions demonstrating that valid governing inputs produce the required conforming behavior.

Positive tests MAY include:

- sufficient authority;
- sufficient evidence;
- valid Rule-State;
- valid context;
- satisfied dependencies;
- admissible action;
- eligible execution;
- valid continuity state; and
- successful recomputation.

**14.17 Negative Testing**

Conformance testing SHALL include negative conditions sufficient to demonstrate that invalid or incomplete governing conditions do not produce improper affirmative outcomes.

Negative testing SHALL include, where applicable:

- missing authority;
- invalid authority;
- out-of-scope authority;
- revoked authority;
- missing evidence;
- invalid evidence;
- insufficient evidence;
- revoked evidence;

- invalid Rule-State;
- superseded Rule-State;
- conflicting rules;
- unresolved ambiguity;
- invalid context;
- unmet dependency;
- expired Trust-State;
- expired eligibility;
- integrity failure;
- serialization failure;
- missed material change;
- missed revocation;
- attempted execution bypass;
- replay mismatch; and
- insufficient recomputation data.

**14.18 Failure-Test Evidence**

A conformance claim SHALL preserve evidence demonstrating that required failure behavior occurred.

Where fail-closed behavior is required, test evidence SHALL demonstrate that execution was prevented when governing conditions were not established.

A claim SHALL NOT rely solely upon successful-path testing.

**14.19 Deterministic Test Results**

Where deterministic behavior is required, repeated tests using equivalent governing conditions SHALL produce equivalent governed outcomes.

Test evidence SHOULD include repeated execution sufficient to establish that results are not dependent upon:

- timing variation;
- implementation randomness;

- hidden state;
- environment-specific behavior; or
- undisclosed discretionary logic.

**14.20 Cross-Implementation Evidence**

Where cross-implementation reproducibility is part of the applicable claim, conformance evidence SHALL demonstrate that independent conforming implementations produce equivalent governed results from equivalent inputs.

Cross-implementation evidence MAY include:

- canonical test vectors;
- recomputation results;
- replay results;
- integrity values;
- equivalence reports; and
- discrepancy reports.

**14.21 Test Vector Requirements**

Where normative test vectors are provided, a conforming implementation SHALL demonstrate the required result for the applicable vectors.

Test vectors SHALL identify, where applicable:

- governed inputs;
- Authority-State;
- evidence;
- Rule-State;
- context;
- expected admissibility outcome;
- expected Trust-State;
- expected eligibility outcome;
- expected canonical representation;

- expected integrity value; and
- expected failure result.

**14.22 Requirement Traceability**

A conformance assessment SHALL maintain traceability between:

- normative requirement;
- implementation control;
- test procedure;
- evidence;
- result; and
- final conformity determination.

Traceability SHALL be sufficient for independent review.

**14.23 Evidence Provenance**

Conformance evidence SHALL preserve provenance sufficient to establish:

- evidence source;
- collection method;
- test environment;
- evaluator;
- test time;
- implementation version;
- test procedure version;
- relevant transformations; and
- verification history.

**14.24 Evidence Integrity**

Conformance evidence SHALL be integrity-protected where alteration could materially affect the conformance result.

Where integrity values are used, the applicable canonicalization and integrity procedures SHALL be identifiable.

**14.25 Assessment Environment**

A conformance claim SHALL identify the assessed implementation environment where environment can materially affect conformity.

The assessment environment MAY include:

- software version;
- configuration;
- policy configuration;
- execution environment;
- relevant dependencies;
- cryptographic configuration;
- deployment architecture; and
- external services.

**14.26 Implementation Version**

The implementation assessed SHALL be version-identifiable.

A materially changed implementation SHALL NOT automatically inherit a prior conformance claim.

Change impact SHALL be evaluated according to the applicable governance process.

**14.27 Configuration-Specific Conformance**

A system MAY conform only under a defined configuration.

Where so, the conformance claim SHALL identify the required configuration.

A claim SHALL NOT imply conformity under configurations that bypass or disable mandatory controls.

**14.28 Dependency-Specific Conformance**

Where conformity depends upon an external dependency, the claim SHALL identify the dependency sufficiently to determine:

- what function it performs;
- what profile requirement depends upon it;
- how its outputs are verified;
- whether its historical state is reproducible where required; and
- what happens if the dependency fails.

**14.29 Evidence Retention**

Conformance evidence SHALL be retained for the period required by the applicable:

- conformance program;
- certification program;
- Rule-State;
- governance policy;
- contractual obligation; or
- regulatory requirement.

The claim SHALL NOT imply evidence availability beyond the declared retention boundary.

**14.30 Claim Expiration**

A conformance claim MAY have a defined expiration or review period.

Where expiration applies:

- the expiration condition SHALL be identifiable;
- expired claims SHALL NOT be represented as current; and
- reassessment or renewal SHALL occur before the claim is restored.

**14.31 Claim Suspension**

A conformance claim SHALL be suspended where the applicable governance or certification program determines that continued conformity can no longer be established.

Suspension MAY result from:

- material implementation change;

- unresolved non-conformity;
- failed reassessment;
- integrity failure;
- evidence loss;
- governance failure; or
- another defined condition.

A suspended claim SHALL NOT be presented as active conformity.

**14.32 Claim Withdrawal**

A conformance claim MAY be withdrawn by the claimant or applicable certification authority.

Withdrawal SHALL be distinguishable from expiration, suspension, and non-conformity.

Withdrawal history SHOULD be preserved where required by the applicable program.

**14.33 Material Change to a Conforming Implementation**

Material change to an implementation SHALL trigger impact assessment where the change can affect conformity.

Material change MAY include:

- evaluation-logic change;
- Rule-State handling change;
- canonicalization change;
- enforcement change;
- continuity change;
- recomputation change;
- dependency change;
- cryptographic change;
- action-class change; or
- conformance-boundary change.

**14.34 Reassessment**

A material change MAY require:

- targeted reassessment;
- partial reassessment;
- full reassessment; or
- no reassessment

depending upon its impact.

The basis for the reassessment decision SHALL be documented.

**14.35 Claim Language**

A public conformance claim SHALL use language sufficient to avoid ambiguity regarding profile and scope.

Preferred claim structure:

**Conforms to Trust-State Standard v[version], Profile [I/II/III], within the declared conformance boundary identified in [record or identifier].**

Equivalent wording MAY be used where the same information remains clearly available.

**14.36 Prohibited Unqualified Claims**

An implementation SHALL NOT make unqualified claims such as:

- “Trust-State compliant”;
- “fully Trust-State compliant”;
- “Trust-State approved”;
- “Trust-State verified”; or
- “Trust-State certified”

unless the associated claim or certification record identifies the applicable Standard version, Conformance Profile, and boundary.

**14.37 Certification Claims**

Where third-party certification exists, the public certification claim SHALL identify or link to a certification record containing:

- certified implementation;
- Standard version;
- Conformance Profile;
- certified boundary;
- issuing certification body;
- certification status;
- issue date;
- expiration or renewal condition where applicable; and
- certification identifier.

**14.38 Certification Mark Use**

Use of a certification mark SHALL be governed by the applicable certification program.

A certification mark SHALL NOT be used:

- outside the certified boundary;
- for an uncertified profile;
- after suspension or expiration;
- for a materially changed implementation not covered by certification; or
- in a manner that implies broader conformity than certified.

**14.39 Self-Declaration**

A system operator MAY self-declare conformance where permitted by the applicable program.

A self-declaration SHALL:

- identify itself as a self-declaration;
- meet the same claim specificity requirements;
- preserve supporting evidence;

- identify the responsible declaring entity; and
- SHALL NOT be represented as third-party certification.

**14.40 Third-Party Assessment**

A third party MAY assess conformance without issuing certification.

Where so, the resulting claim SHALL distinguish:

- assessment;
- verification;
- certification; and
- accreditation

where those terms have different meanings under the applicable program.

**14.41 No Broader Compliance Implication**

A Trust-State conformance claim SHALL NOT imply conformity with an external law, regulation, standard, contract, or certification program unless separately established.

A statement that TSS conformance is relevant to another framework SHALL be distinguishable from a claim of compliance with that framework.

**14.42 No Authority Endorsement Implication**

A conformance claim SHALL NOT imply that the Trust-State Standard, VTI Foundation, or a certification body endorses:

- the substantive authority granted by the implementation;
- the policies encoded in the Rule-State;
- the evidence sources selected;
- the consequences permitted; or
- the business, legal, ethical, or regulatory correctness of the implementation.

Conformance concerns the governed mechanism and declared capability.

**14.43 No Outcome Guarantee**

A conformance claim SHALL NOT state or imply that TSS conformity guarantees a desired consequence.

Conformance establishes that declared governing conditions are evaluated and governed according to the applicable requirements.

It does not guarantee the substantive outcome of every consequence-bearing action.

**14.44 Machine-Readable Claim**

A conforming implementation MAY provide a machine-readable conformance claim.

Where provided, the claim SHOULD include fields representing:

- Standard version;
- Conformance Profile;
- implementation identifier;
- conformance boundary identifier;
- assessment status;
- assessment date;
- applicable specification versions;
- certification identifier where applicable;
- evidence-reference identifier; and
- claim integrity value.

**14.45 Claim Integrity**

A machine-verifiable conformance claim SHALL be integrity-protected where required by the applicable certification or conformance program.

Integrity protection SHALL enable detection of material alteration to:

- Standard version;
- profile;
- boundary;
- status;

- implementation identity; and
- associated certification or assessment identifiers.

**14.46 Claim Verification**

Where machine-verifiable claims are supported, an independent verifier SHALL be able to determine:

- who or what made the claim;
- what implementation the claim applies to;
- what Standard version applies;
- what profile applies;
- what boundary applies;
- whether the claim is current;
- whether integrity verifies; and
- whether supporting evidence or certification record is available as required.

**14.47 Conformance Record**

A conformance assessment SHALL produce a conformance record sufficient to support the associated claim.

The record SHALL include or reference:

- claim elements;
- requirement mapping;
- assessment evidence;
- test results;
- non-conformities;
- limitations;
- exceptions;
- assessor identity where applicable;
- final determination; and
- integrity information.

**14.48 Non-Conformity Record**

Where a mandatory requirement is not satisfied, the conformance record SHALL identify the non-conformity.

The record SHOULD identify:

- affected requirement;
- evidence;
- severity or classification where used;
- remediation status;
- reassessment requirement; and
- effect upon the conformance claim.

**14.49 Conformance Evidence Package**

A conformance evidence package MAY aggregate the evidence supporting a claim.

Where used, the package SHALL preserve sufficient structure to distinguish:

- normative requirement;
- evidence item;
- test result;
- determination;
- limitation;
- non-conformity;
- profile; and
- conformance boundary.

**14.50 Independent Verification of Claim**

A conformance claim SHALL be independently reviewable to the level required by the applicable program.

An independent reviewer SHALL be able to determine:

- whether the claimed profile is supported;

- whether the boundary is accurately stated;
- whether required evidence exists;
- whether mandatory requirements were tested;
- whether material failures were disclosed;
- whether claim status is accurate; and
- whether the claim exceeds the demonstrated capability.

**14.51 False or Misleading Claim**

A claim SHALL be considered false or misleading where it materially:

- overstates Conformance Profile;
- omits a material boundary limitation;
- represents conditional status as full conformity;
- represents self-declaration as certification;
- uses an expired or suspended certification;
- represents untested action classes as conforming;
- implies regulatory compliance without separate basis; or
- otherwise exceeds the demonstrated conformance state.

**14.52 Conformance Requirement**

A system or organization conforms to this section when its Trust-State conformance claims are:

- version-specific;
- profile-specific;
- boundary-specific;
- evidence-supported;
- traceable;
- integrity-protected where required;
- independently reviewable; and
- limited to the capability actually demonstrated.

### 14.53 Canonical Statement

**Conformance is demonstrated capability, not marketing language.**

A valid claim identifies what conforms, to which version, under which profile, within which boundary, and on the basis of what evidence.

No Trust-State claim extends beyond the evidence that supports it.

## Section 15 — Certification and Governance

### 15.0 Purpose

This section defines the normative requirements governing certification, certification records, assessment authority, certification status, certification lifecycle, and governance of the Trust-State Standard certification framework.

Certification is a formal determination that an identified implementation conforms to a specified Trust-State Standard release, Conformance Profile, and conformance boundary.

This section is normative.

### 15.1 Certification Principle

Certification SHALL be based upon demonstrated conformity.

Certification SHALL NOT substitute for conformance testing.

A certification decision SHALL be supported by evidence sufficient to establish that:

- the applicable Standard version was identified;
- the applicable Conformance Profile was identified;
- the conformance boundary was defined;
- applicable normative requirements were evaluated;
- required conformance evidence was reviewed;
- material non-conformities were resolved or otherwise governed;
- the assessed implementation was version-identifiable; and
- the certification decision is independently traceable.

### 15.2 Conformance and Certification Separation

Conformance and certification SHALL remain distinct.

Conformance is satisfaction of the normative requirements applicable to a declared Conformance Profile and boundary.

Certification is a formal third-party determination regarding that conformity.

An implementation MAY conform without being certified unless another governing requirement mandates certification.

Certification SHALL NOT create conformity where conformity has not been demonstrated.

**15.3 Certification Scope**

A certification SHALL apply only to the:

- identified implementation;
- identified implementation version;
- applicable Trust-State Standard release;
- applicable Conformance Profile;
- declared conformance boundary;
- assessed action class or action classes;
- applicable configuration;
- applicable dependencies; and
- certification validity period or lifecycle state.

Certification SHALL NOT imply conformity outside that scope.

**15.4 Certification Profile**

A certification SHALL identify one of the Trust-State Standard Conformance Profiles defined in Section 13.

Certification MAY be issued for:

- Profile I — Deterministic Admissibility;
- Profile II — Deterministic Enforcement; or
- Profile III — Deterministic Continuity and Recomputation.

A certification SHALL NOT imply capability associated with a higher profile unless the higher profile has been independently demonstrated.

**15.5 Certification Boundary**

The certification boundary SHALL identify the scope within which conformity has been assessed.

The boundary SHALL identify, where applicable:

- system or implementation;

- component set;
- action classes;
- actor classes;
- resources;
- Authority-State boundary;
- evidence boundary;
- Rule-State boundary;
- contextual boundary;
- execution boundary;
- continuity boundary;
- replay boundary;
- retention boundary;
- external dependencies;
- geographic or jurisdictional limitations; and
- applicable configurations.

**15.6 Certification Authority**

Certification SHALL be issued only by an entity authorized under the applicable Trust-State certification governance framework.

Certification authority MAY be exercised by:

- VTI Foundation;
- an authorized certification body;
- an accredited conformity assessment body;
- another organization formally recognized under the applicable certification program.

Authorization to certify SHALL be documented.

**15.7 Certification Body Independence**

A certification body SHALL maintain sufficient independence to make certification determinations based upon conformity evidence rather than commercial, operational, or implementation interests.

Where the certification body also provides implementation, consulting, testing, or advisory services, applicable governance controls SHALL address conflicts of interest.

**15.8 Separation of Roles**

The certification framework SHOULD distinguish, where applicable, between:

- Standard stewardship;
- implementation;
- testing;
- conformity assessment;
- certification decision;
- accreditation;
- appeals;
- certification mark administration; and
- enforcement of certification-program rules.

One organization MAY perform more than one function only where applicable independence and conflict controls are preserved.

**15.9 Assessment Authority**

A conformity assessment used for certification SHALL be performed by a qualified assessor, assessment body, test system, or other authorized evaluation mechanism.

The certification program SHALL define qualification requirements applicable to assessors.

Qualification MAY include:

- technical competency;
- standards competency;
- conformance-testing competency;
- independence requirements;

- conflict-of-interest controls;
- domain competency; and
- certification-program training.

**15.10 Certification Evidence**

A certification decision SHALL be supported by a conformance evidence package sufficient to demonstrate satisfaction of the applicable profile requirements.

The evidence package SHALL include or reference:

- conformance claim;
- conformance boundary;
- requirement mapping;
- positive test results;
- negative test results;
- profile-specific evidence;
- implementation identity;
- applicable configuration;
- applicable dependencies;
- non-conformity records;
- remediation evidence;
- assessor determination; and
- integrity information where required.

**15.11 Certification Decision**

A certification decision SHALL be distinct from evidence collection and test execution where required by the applicable certification program.

The decision SHALL identify:

- certification outcome;
- Standard version;

- Conformance Profile;
- certified implementation;
- certified boundary;
- material limitations;
- unresolved conditions where permitted;
- certification status;
- decision date;
- certification identifier; and
- applicable validity or renewal conditions.

**15.12 Certification Decision Basis**

A certification decision SHALL be based upon objectively reviewable evidence.

A certification SHALL NOT be issued solely on the basis of:

- vendor assertion;
- architectural intent;
- reputation;
- policy statement;
- marketing material;
- self-attestation;
- documentation without verification; or
- subjective confidence.

**15.13 Certification Status**

A certification record SHALL identify certification status.

Status MAY include:

- ACTIVE;
- CONDITIONAL;
- SUSPENDED;

- EXPIRED;
- WITHDRAWN;
- REVOKED;
- PENDING_RENEWAL; or
- another objectively defined certification-program state.

Status criteria SHALL be governed by the applicable certification program.

**15.14 Active Certification**

An ACTIVE certification SHALL indicate that the certified implementation remains within the certified Standard version, profile, boundary, and validity conditions.

Active certification SHALL NOT imply that no changes have occurred.

It indicates that changes affecting certification have been governed according to the applicable certification lifecycle requirements.

**15.15 Conditional Certification**

Conditional certification MAY be used only where expressly permitted by the applicable certification program.

Where conditional certification is permitted, the certification record SHALL identify:

- unresolved condition;
- permitted certification scope;
- required remediation;
- temporal limit;
- restrictions;
- review condition; and
- condition resulting in suspension or withdrawal.

Conditional certification SHALL NOT be used for an unresolved failure that materially defeats a mandatory requirement of the claimed profile.

**15.16 Certification Suspension**

Certification SHALL be suspended where continued certified conformity cannot be established but immediate revocation is not required.

Suspension MAY result from:

- material implementation change;
- unresolved non-conformity;
- failed surveillance;
- failed reassessment;
- evidence integrity failure;
- certification-program violation;
- inability to verify continued conformity; or
- another defined condition.

A suspended certification SHALL NOT be represented as active.

**15.17 Certification Expiration**

A certification MAY have a defined validity period.

Where expiration applies:

- the expiration date or condition SHALL be identifiable;
- expired certification SHALL NOT be represented as active;
- continued certification SHALL require renewal or reassessment as defined by the certification program.

**15.18 Certification Withdrawal**

Certification MAY be withdrawn by:

- the certification holder;
- the certification body;
- the certification program authority; or
- another authorized party under defined governance conditions.

Withdrawal SHALL be distinguishable from suspension, expiration, and revocation.

**15.19 Certification Revocation**

Certification SHALL be revoked where the applicable certification program determines that certification is no longer supportable due to material failure.

Revocation MAY result from:

- material false statement;
- falsified evidence;
- intentional boundary misrepresentation;
- unauthorized certification-mark use;
- unremediated critical non-conformity;
- material integrity failure;
- certification-program abuse; or
- another defined revocation condition.

Revocation history SHALL be preserved.

**15.20 Certification Lifecycle**

The certification program SHALL define the certification lifecycle.

The lifecycle MAY include:

- application;
- scope definition;
- assessment;
- testing;
- evidence review;
- non-conformity resolution;
- certification decision;
- issuance;
- surveillance;

- change review;
- reassessment;
- renewal;
- suspension;
- withdrawal;
- revocation; and
- closure.

**15.21 Surveillance**

A certification program MAY require ongoing surveillance.

Surveillance MAY evaluate:

- continued implementation conformity;
- changes to evaluation logic;
- changes to Rule-State handling;
- changes to Authority-State handling;
- changes to evidence handling;
- execution-enforcement integrity;
- continuity behavior;
- replay capability;
- material dependency changes;
- certification-mark use; and
- continued evidence availability.

**15.22 Material Change**

A certified implementation SHALL identify material changes capable of affecting the certified conformance state where required by the certification program.

Material change MAY include:

- implementation change;

- architecture change;
- evaluation-logic change;
- enforcement change;
- canonicalization change;
- cryptographic change;
- Rule-State handling change;
- evidence handling change;
- continuity change;
- replay change;
- dependency change;
- configuration change;
- action-class expansion; or
- conformance-boundary change.

**15.23 Change Impact Assessment**

A material change SHALL be subject to impact assessment where required by the certification program.

The impact assessment SHALL determine whether the change requires:

- no certification action;
- documentation update;
- targeted testing;
- partial reassessment;
- full reassessment;
- certification suspension; or
- new certification.

**15.24 Certification Renewal**

Certification renewal SHALL require sufficient evidence to establish continued conformity.

Renewal MAY include:

- reassessment of unchanged controls;
- reassessment of changed controls;
- updated negative testing;
- updated replay testing;
- review of material changes;
- review of non-conformities;
- review of certification-mark use; and
- confirmation of the certification boundary.

**15.25 Certification Record**

Each certification SHALL have an associated certification record.

The record SHALL identify at minimum:

- certification identifier;
- certified implementation;
- implementation version;
- certification holder;
- Trust-State Standard version;
- Conformance Profile;
- certification boundary;
- certification body;
- certification status;
- issue date;
- expiration or renewal condition where applicable;
- material limitations;
- applicable normative annexes; and
- record integrity information where required.

**15.26 Certification Record Integrity**

A machine-verifiable certification record SHALL be integrity-protected.

Integrity protection SHALL enable detection of material alteration to:

- certification identifier;
- implementation identity;
- Standard version;
- Conformance Profile;
- certification boundary;
- certification status;
- issue date;
- expiration state; and
- certification body identity.

**15.27 Public Verification**

Where certification is publicly represented, the certification program SHOULD provide a method by which certification status can be independently verified.

Verification MAY be supported through:

- certification registry;
- machine-readable certification record;
- verification service;
- signed artifact;
- published certification record; or
- another authoritative mechanism.

**15.28 Certification Mark**

A certification program MAY authorize use of a certification mark.

Use of a certification mark SHALL correspond to an active certification and SHALL be limited to the certified:

- implementation;
- profile;
- boundary;
- version;
- configuration; and
- validity period.

**15.29 Certification Mark Profile Identification**

Where certification marks are used, associated records SHALL identify the applicable Conformance Profile.

A certification mark SHALL NOT imply that all Trust-State Standard profiles have been certified unless such certification has been separately established.

**15.30 Certification Mark Misuse**

Certification-mark misuse SHALL include, where applicable:

- use without certification;
- use outside certified scope;
- use for an uncertified profile;
- use after suspension;
- use after expiration;
- use after withdrawal;
- use after revocation;
- use on a materially changed implementation not covered by certification; or
- use implying regulatory approval not actually established.

**15.31 Certification and Regulatory Status**

Trust-State certification SHALL NOT, by itself, establish compliance with:

- law;

- regulation;
- regulatory guidance;
- safety requirement;
- privacy requirement;
- cybersecurity requirement;
- procurement rule; or
- another external framework.

Where an external framework recognizes or references Trust-State certification, that relationship SHALL be separately established.

**15.32 Certification and Legal Authority**

Certification SHALL NOT confer legal authority upon an actor or system.

Certification verifies conformity of the governed mechanism within the certified boundary.

Authority remains governed through the Authority-State and applicable external authority sources.

**15.33 Certification and Policy Validity**

Certification SHALL NOT constitute approval of the substantive policies encoded within the Rule-State.

Certification evaluates whether applicable policies are identified, governed, evaluated, enforced, and preserved according to the applicable profile.

It does not determine whether those policies are substantively desirable, lawful, ethical, or optimal.

**15.34 Certification and Consequence**

Certification SHALL NOT constitute a guarantee of desired consequence.

Certification demonstrates conformity of the governed execution mechanism.

It does not guarantee that every action will produce a desired outcome.

**15.35 Certification Governance Authority**

Governance of the Trust-State Standard certification framework SHALL be maintained under the applicable standards and certification governance authority designated by VTI Foundation, Inc.

Certification governance SHALL include, where applicable:

- certification-program rules;
- authorization of certification bodies;
- assessor qualification criteria;
- certification-mark rules;
- registry requirements;
- appeals procedures;
- suspension procedures;
- withdrawal procedures;
- revocation procedures;
- renewal requirements; and
- change-control requirements.

**15.36 Standards Stewardship and Certification Separation**

Standards stewardship and certification administration SHALL remain distinguishable functions.

Standards stewardship governs:

- normative publication;
- versioning;
- terminology;
- maintenance;
- interpretation governance;
- annex status; and
- change control.

Certification administration governs:

- conformity assessment;
- certification decision;
- certification status;
- certification records;

- certification-mark use; and
- certification lifecycle.

**15.37 Commercial Implementation Separation**

A commercial implementation provider SHALL NOT control normative Trust-State Standard requirements solely through product implementation.

The normative Standard SHALL remain independent from any specific commercial product, platform, protocol, or vendor implementation.

**15.38 Certification Conflict of Interest**

The certification program SHALL define controls addressing material conflicts of interest.

Controls MAY include:

- disclosure;
- assessor separation;
- certification-decision separation;
- recusal;
- independent review;
- external assessment; or
- other governed mechanisms.

**15.39 Appeals**

The certification program SHOULD provide a governed appeals process.

An appeal process SHOULD identify:

- eligible appellant;
- appealable decision;
- filing period;
- evidence requirements;
- reviewer independence;

- decision procedure;
- possible outcomes; and
- final disposition.

An appeal SHALL NOT automatically restore suspended or revoked certification unless the applicable process expressly provides otherwise.

**15.40 Complaints**

The certification program MAY define a complaints process for allegations concerning:

- certification misuse;
- certification-mark misuse;
- non-conforming certified implementation;
- assessor conduct;
- certification-body conduct;
- false certification claims; or
- other program-governance matters.

Complaints SHALL be distinguished from formal appeals.

**15.41 Corrective Action**

Where a remediable non-conformity is identified, the certification program MAY permit corrective action.

Corrective action SHALL identify:

- affected requirement;
- remediation;
- remediation evidence;
- verification procedure;
- completion date;
- reassessment requirement; and
- certification-status effect.

**15.42 Certification Audit Trail**

The certification process SHALL preserve an audit trail sufficient to establish:

- application;
- assessed boundary;
- assessor activity;
- test results;
- evidence reviewed;
- non-conformities;
- corrective actions;
- certification decision;
- subsequent material changes;
- surveillance;
- renewal;
- suspension;
- withdrawal; and
- revocation.

**15.43 Certification Retention**

Certification records and supporting evidence SHALL be retained according to the applicable certification-program retention policy.

Retention SHALL support verification of certification history for the defined retention period.

**15.44 Certification Transparency**

Public certification information SHOULD be sufficient to prevent misleading claims while protecting information not required for public disclosure.

Public certification information SHOULD include:

- certification holder;
- certified implementation;

- Standard version;
- Conformance Profile;
- certified boundary summary;
- certification status;
- certification body;
- issue date; and
- certification identifier.

**15.45 Confidential Certification Evidence**

Certification evidence MAY contain confidential, proprietary, security-sensitive, personal, or regulated information.

The certification program SHALL define controls governing:

- access;
- disclosure;
- retention;
- handling;
- verification;
- assessor access; and
- destruction where applicable.

Confidentiality SHALL NOT eliminate the requirement for sufficient evidence to support certification.

**15.46 Certification Portability**

Certification SHALL apply to the certified implementation and boundary.

Certification SHALL NOT automatically transfer to:

- another product;
- another deployment;
- another organization;
- another implementation;

- another action class;
- another Rule-State;
- another configuration; or
- another jurisdiction

where material governing conditions differ.

**15.47 Certification Reuse**

Prior certification evidence MAY be reused where the certification program determines that the evidence remains:

- applicable;
- current;
- valid;
- integrity-preserved;
- within scope; and
- relevant to the current assessment.

Reuse SHALL NOT substitute for reassessment of materially changed conditions.

**15.48 Multi-Profile Certification**

An implementation MAY hold certification for more than one Conformance Profile or more than one conformance boundary.

Each certified profile and boundary SHALL remain independently identifiable.

**15.49 Certification Inheritance**

A higher-profile certification MAY satisfy lower-profile requirements where:

- the higher profile incorporates the lower-profile requirements;
- the same certified boundary applies;
- no material profile-specific exclusion exists; and
- the certification program permits inheritance.

Inheritance SHALL be documented in the certification record.

**15.50 Certification Non-Conformity**

Certification non-conformity exists where:

- mandatory profile requirements are not satisfied;
- certification evidence is insufficient;
- the certified boundary is misrepresented;
- required controls have been disabled;
- material changes were not governed as required;
- certification-mark conditions are violated;
- required continuity cannot be demonstrated;
- required replay cannot be demonstrated; or
- another certification-program requirement is materially unsatisfied.

**15.51 Governance of Certification Changes**

Material changes to certification-program rules SHALL be version-controlled.

Certification governance changes SHALL identify:

- prior rule;
- revised rule;
- effective date;
- applicability;
- transition requirements;
- effect on existing certifications; and
- governance approval.

A later certification-program rule SHALL NOT silently alter the status of an existing certification unless the governing transition rules expressly provide such effect.

### 15.52 Conformance Requirement

A certification program conforms to this section when certification is:

- based upon demonstrated conformity;
- version-specific;
- profile-specific;
- boundary-specific;
- evidence-supported;
- independently reviewable;
- governed through a defined lifecycle;
- protected against misleading use;
- distinguishable from regulatory approval;
- separated from substantive policy endorsement; and
- maintained under transparent governance controls.

### 15.53 Canonical Statement

**Conformance establishes demonstrated capability.**

Certification establishes that an authorized third party has evaluated that capability within a defined profile and boundary.

Certification does not create authority, validate policy, guarantee consequence, or establish regulatory compliance beyond what has been separately demonstrated.

## Section 16 — Security Considerations

### 16.0 Purpose

This section defines the normative security considerations applicable to Trust-State implementations.

Security under the Trust-State Standard is concerned with preserving the integrity, authenticity, availability, traceability, and enforceability of the governed conditions upon which admissibility, execution eligibility, continuity, recomputation, and certification depend.

This section is normative.

### 16.1 Security Principle

A conforming system SHALL protect against conditions that could cause a consequence-bearing action to be treated as admissible, eligible, valid, continuous, or conforming when the governing conditions do not support that result.

Security controls SHALL protect the integrity of:

- Authority-State;
- evidence;
- Rule-State;
- context;
- admissibility determinations;
- Trust-State;
- execution eligibility;
- enforcement mechanisms;
- continuity state;
- revocation state;
- canonical representations;
- integrity values;
- replay state; and
- conformance evidence.

**16.2 Security Boundary**

A conforming implementation SHALL define the security boundary applicable to its declared conformance scope.

The boundary SHALL identify, where applicable:

- governed components;
- trust dependencies;
- execution gates;
- evidence stores;
- Rule-State sources;
- Authority-State sources;
- canonicalization components;
- integrity mechanisms;
- replay stores;
- external services;
- administrative interfaces; and
- certification-relevant interfaces.

A security claim SHALL NOT extend beyond the declared security boundary.

**16.3 Threat Model**

A conforming system SHALL identify threats capable of materially altering governed evaluation or execution behavior.

Threats SHALL include, where applicable:

- unauthorized Authority-State creation;
- authority escalation;
- authority substitution;
- delegation forgery;
- evidence fabrication;
- evidence substitution;

- evidence tampering;
- evidence suppression;
- Rule-State modification;
- Rule-State substitution;
- context falsification;
- revocation suppression;
- replay of stale authorization;
- execution-gate bypass;
- determination tampering;
- canonicalization manipulation;
- integrity-value substitution;
- hidden-state injection;
- dependency compromise;
- privileged administrative misuse; and
- conformance-evidence falsification.

**16.4 Authority-State Security**

A conforming system SHALL protect Authority-State against unauthorized:

- creation;
- modification;
- delegation;
- scope expansion;
- constraint removal;
- substitution;
- suspension reversal;
- revocation reversal; and
- deletion where deletion would impair verification.

Authority-State security SHALL preserve the distinction between valid authority and technical possession of access or capability.

**16.5 Delegation Security**

Where delegated authority is supported, the system SHALL protect delegation lineage against:

- forged delegation;
- unauthorized sub-delegation;
- scope expansion;
- constraint removal;
- temporal extension;
- subject substitution;
- delegation replay; and
- revocation suppression.

A delegation chain that cannot be integrity-verified SHALL NOT be treated as sufficient authority where verified lineage is required.

**16.6 Evidence Security**

Evidence relied upon in admissibility determinations SHALL be protected against:

- fabrication;
- unauthorized modification;
- substitution;
- deletion;
- truncation;
- stale reuse;
- provenance loss;
- invalid transformation;
- revocation suppression; and
- scope misrepresentation.

Security controls SHALL be proportionate to the materiality of the evidence within the governing determination.

**16.7 Evidence Source Compromise**

A conforming system SHALL account for the possibility that a recognized evidence source may become compromised.

Where source compromise is material, the applicable Rule-State SHALL define or support determination of:

- affected evidence;
- validity impact;
- revocation impact;
- continuity impact;
- required re-evaluation;
- required replacement evidence; and
- affected execution eligibility.

Prior recognition of a source SHALL NOT create permanent evidentiary trust.

**16.8 Rule-State Security**

A conforming system SHALL protect Rule-State against unauthorized:

- modification;
- replacement;
- version rollback;
- deletion;
- precedence alteration;
- conflict-resolution alteration;
- constraint removal;
- effective-date manipulation; and
- governance-source substitution.

Material Rule-State changes SHALL be version-identifiable and provenance-preserved.

**16.9 Rule-State Rollback**

A system SHALL prevent unauthorized rollback to a prior Rule-State where rollback could alter admissibility or execution eligibility.

Authorized rollback MAY occur only through governed change control.

Rollback events SHALL preserve:

- prior active Rule-State;
- restored Rule-State;
- authorizing authority;
- effective time;
- justification; and
- affected boundary.

**16.10 Context Security**

Where context materially affects admissibility, the system SHALL protect contextual inputs against falsification or substitution.

Protected context MAY include:

- time;
- location;
- jurisdiction;
- actor state;
- resource state;
- system state;
- dependency state;
- transaction state; and
- environmental conditions.

A context value SHALL NOT be treated as reliable solely because it originates inside the implementation.

**16.11 Time Security**

Where temporal state materially affects authority, evidence, Rule-State, expiration, revocation, or eligibility, the system SHALL protect the integrity of relevant time sources and time representations.

The system SHALL account for threats including:

- clock manipulation;
- timestamp substitution;
- timezone ambiguity;
- stale timestamp reuse;
- expiration bypass; and
- delayed revocation processing.

**16.12 Determination Security**

Admissibility, Trust-State, continuity, and execution eligibility determinations SHALL be protected against unauthorized modification.

Security controls SHALL prevent or detect:

- outcome substitution;
- action substitution;
- input substitution;
- Rule-State substitution;
- Authority-State substitution;
- evidence-set substitution;
- context substitution; and
- integrity-value substitution.

**16.13 Action Binding Security**

A determination SHALL be protected against use for a materially different action than the action for which it was derived.

The system SHALL prevent or detect:

- action identifier substitution;

- resource substitution;
- actor substitution;
- target substitution;
- scope expansion; and
- transaction substitution.

A valid determination SHALL NOT be portable to an unrelated consequence-bearing action unless the Rule-State expressly permits such reuse.

**16.14 Execution-Gate Security**

Where Profile II or Profile III enforcement applies, the execution gate SHALL be protected against bypass.

A conforming system SHALL prevent or detect:

- alternate execution paths;
- direct backend invocation;
- privileged bypass;
- stale eligibility reuse;
- forged eligibility;
- gate disabling;
- policy-enforcement suppression; and
- execution outside the governed pathway.

An execution path capable of bypassing required enforcement SHALL constitute a security-relevant condition.

**16.15 Fail-Closed Security**

Where fail-closed behavior is required, security failure SHALL NOT silently convert an indeterminate state into permission to execute.

Examples include:

- unavailable Authority-State source;
- unavailable evidence source;
- Rule-State retrieval failure;

- integrity-verification failure;
- revocation-service failure;
- context-verification failure; and
- dependency failure.

**16.16 Privileged Access**

Privileged access capable of altering Trust-State governance SHALL itself be governed.

Privileged operations MAY include:

- modifying Rule-State;
- issuing Authority-State;
- revoking authority;
- changing evidence state;
- overriding eligibility;
- disabling enforcement;
- altering canonicalization configuration;
- modifying audit evidence; and
- changing certification-relevant controls.

Privileged capability SHALL NOT imply unrestricted authority to perform such operations.

**16.17 Administrative Override**

Administrative override SHALL NOT provide an ungoverned bypass around admissibility or execution requirements.

Where override is permitted, it SHALL satisfy the override requirements defined elsewhere in this Standard and SHALL preserve:

- authorized actor;
- authority basis;
- scope;
- justification;

- evidence;
- effective time;
- resulting state; and
- provenance.

**16.18 Separation of Duties**

Where separation of duties is required by the Rule-State, the system SHALL enforce the applicable role and authority separation.

Separation MAY apply to:

- authority issuance;
- evidence approval;
- Rule-State modification;
- admissibility review;
- execution approval;
- revocation;
- certification assessment; and
- certification decision.

A single actor SHALL NOT perform conflicting functions where the governing Rule-State prohibits such combination.

**16.19 Integrity Mechanism Security**

Integrity mechanisms used under Section 11 SHALL be implemented so that an attacker cannot materially alter governed content without detection to the level required by the applicable Conformance Profile.

Where cryptographic mechanisms are used, the implementation SHALL protect:

- key material;
- algorithm selection;
- algorithm parameters;
- key rotation;

- verification keys;
- signing authority;
- trust anchors; and
- algorithm identifiers.

**16.20 Key Compromise**

Where cryptographic keys materially support authority, evidence, integrity, or certification, the implementation SHALL define handling for key compromise.

Key compromise handling MAY include:

- suspension;
- revocation;
- replacement;
- re-signing;
- revalidation;
- impact analysis;
- continuity re-evaluation; and
- notification.

Compromised key material SHALL NOT remain a valid basis for new governed assertions once compromise is established.

**16.21 Algorithm Agility**

A conforming implementation SHOULD support governed transition away from cryptographic algorithms that become unsuitable.

Algorithm transition SHALL preserve, where applicable:

- prior verification capability;
- historical artifact verification;
- version identity;
- migration provenance;
- affected scope; and

- transition integrity.

A new algorithm SHALL NOT silently invalidate historical evidence without defined transition rules.

**16.22 Canonicalization Security**

Canonicalization procedures SHALL be protected against manipulation capable of producing inconsistent representations from equivalent governed inputs.

Threats SHALL include, where applicable:

- field omission;
- field injection;
- ordering manipulation;
- encoding ambiguity;
- numeric ambiguity;
- timestamp ambiguity;
- duplicate-key handling;
- Unicode normalization differences;
- parser differential behavior; and
- external-reference substitution.

**16.23 Parser and Representation Differential**

Where multiple implementations parse or serialize the same governed object, the applicable canonicalization procedure SHALL minimize ambiguity capable of causing divergent evaluation.

A conforming system SHALL account for differential interpretation where such differences could affect:

- integrity value;
- Rule-State;
- evidence meaning;
- Authority-State;
- admissibility; or
- replay.

**16.24 Replay Attack Prevention**

A conforming system SHALL distinguish legitimate historical replay from unauthorized reuse of prior eligibility, evidence, authority, or execution artifacts.

Where replay of a prior execution could produce unintended consequence, controls SHALL prevent unauthorized duplicate execution.

Controls MAY include:

- nonce;
- transaction identity;
- sequence number;
- freshness condition;
- single-use eligibility;
- expiry;
- state-transition guard; or
- another deterministic replay-prevention mechanism.

**16.25 Stale-State Attack**

A conforming system SHALL prevent or detect reliance upon stale governing state where current state is required.

Stale-state threats MAY include:

- expired Authority-State;
- revoked authority;
- superseded Rule-State;
- expired evidence;
- stale dependency state;
- expired eligibility;
- stale context; and
- outdated revocation state.

**16.26 Revocation Suppression**

A conforming system SHALL protect against suppression, delay, or concealment of material revocation information.

Where revocation state is required before execution, inability to determine required revocation state SHALL be handled according to the applicable fail-closed rule.

**16.27 Rollback and Downgrade Attack**

A conforming system SHALL prevent unauthorized downgrade to:

- weaker Rule-State;
- weaker enforcement mode;
- weaker Conformance Profile behavior;
- obsolete canonicalization version;
- obsolete integrity algorithm;
- prior Authority-State; or
- another state that could improperly increase execution eligibility.

Authorized downgrade SHALL itself be governed and provenance-preserved.

**16.28 Dependency Security**

External dependencies capable of materially affecting governed determinations SHALL be identified and governed.

The implementation SHALL account for threats including:

- dependency unavailability;
- dependency compromise;
- stale dependency state;
- response substitution;
- dependency impersonation;
- undocumented dependency behavior; and
- historical-state loss.

**16.29 Hidden-State Security**

Material hidden state capable of altering governed outcomes SHALL be prohibited where independent recomputation is required.

State affecting admissibility, eligibility, continuity, or replay SHALL be represented or reproducibly available within the applicable governed boundary.

**16.30 Probabilistic Component Security**

Where probabilistic components provide governed inputs, the system SHALL protect against manipulation of those inputs and preserve the deterministic controls governing their use.

Security considerations MAY include:

- model substitution;
- threshold manipulation;
- prompt or input manipulation;
- output substitution;
- model-version change;
- unbounded uncertainty;
- unrecorded output regeneration; and
- provenance loss.

A probabilistic component SHALL NOT be permitted to silently alter deterministic governing logic.

**16.31 Autonomous-Agent Security**

Autonomous operation SHALL NOT weaken the security requirements applicable to governed execution.

A conforming autonomous implementation SHALL account for threats including:

- goal substitution;
- instruction injection;
- delegated-authority expansion;
- tool misuse;
- credential misuse;
- unauthorized chained action;

- persistence of stale authority;
- uncontrolled sub-agent delegation; and
- execution outside declared scope.

**16.32 Delegation to Machine Actors**

Where authority is delegated to a machine actor, the delegation SHALL be protected against:

- scope expansion;
- identity substitution;
- persistence beyond validity;
- unauthorized transfer;
- unauthorized sub-delegation; and
- revocation failure.

Machine actors SHALL NOT infer additional authority from technical capability.

**16.33 Chained-Action Security**

Where one governed action can initiate another consequence-bearing action, the system SHALL prevent an attacker from using a valid upstream determination to bypass downstream admissibility requirements.

Each downstream action SHALL remain subject to the applicable Rule-State and Conformance Profile.

**16.34 Confused-Deputy Protection**

A conforming system SHALL account for conditions in which an actor with valid authority is induced to exercise that authority on behalf of an actor that lacks sufficient authority.

Controls SHOULD bind:

- requesting actor;
- executing actor;
- action;
- resource;
- purpose;

- scope; and
- applicable Authority-State.

**16.35 Cross-Tenant and Cross-Boundary Isolation**

Where multiple tenants, organizations, jurisdictions, or authority domains share infrastructure, the system SHALL prevent governing state from one boundary from being improperly applied to another.

Isolation SHALL protect against:

- Authority-State leakage;
- evidence leakage;
- Rule-State crossover;
- context crossover;
- eligibility reuse;
- artifact substitution; and
- replay across boundaries.

**16.36 Confidentiality**

The Trust-State Standard primarily requires integrity and verifiability of governed state.

Confidentiality SHALL be protected where disclosure of governed information could create:

- security risk;
- privacy harm;
- legal violation;
- commercial harm;
- credential compromise; or
- other material consequence.

Confidentiality controls SHALL NOT prevent required independent verification where verification is mandatory.

**16.37 Data Minimization**

A conforming implementation SHOULD minimize retention and exposure of governed information not required for:

- admissibility;
- enforcement;
- continuity;
- recomputation;
- certification;
- legal obligations; or
- other declared governing requirements.

Canonical representation SHALL NOT require unnecessary inclusion of unrelated sensitive information.

**16.38 Availability**

Where availability of governing inputs is necessary to establish admissibility or eligibility, the implementation SHALL define failure behavior for unavailable inputs.

Availability failure SHALL NOT create affirmative authority, evidence sufficiency, or execution eligibility.

**16.39 Denial-of-Service and Fail-Closed Interaction**

A conforming implementation SHOULD account for the possibility that fail-closed behavior can be intentionally triggered to deny legitimate execution.

Mitigation MAY include:

- redundant authoritative sources;
- cached integrity-protected state within defined freshness bounds;
- alternate verification paths;
- degraded modes governed by Rule-State; and
- emergency procedures.

Availability mitigation SHALL NOT silently convert fail-closed requirements into fail-open behavior.

**16.40 Audit and Evidence Protection**

Security-relevant records SHALL be protected against unauthorized:

- deletion;
- modification;
- reordering;
- timestamp alteration;
- substitution; and
- selective suppression

where such actions could impair independent verification or conformance assessment.

**16.41 Security Event Provenance**

Material security events affecting governed state SHALL preserve sufficient provenance to identify:

- affected state;
- event type;
- detection time;
- effective time where applicable;
- affected boundary;
- response;
- remediation; and
- resulting continuity or conformance impact.

**16.42 Security Incident and Continuity**

A security incident affecting material governing conditions SHALL trigger continuity evaluation where required.

The system SHALL determine whether affected:

- Authority-State;
- evidence;
- Rule-State;

- Trust-State;
- execution eligibility;
- replay data; or
- conformance evidence

remain valid.

**16.43 Security Incident and Certification**

A security incident affecting a certified implementation MAY constitute a material change under Section 15.

The applicable certification program SHALL determine whether the incident requires:

- notification;
- impact assessment;
- targeted reassessment;
- suspension;
- corrective action;
- renewal; or
- revocation.

**16.44 Security Testing**

Conformance testing SHALL include security-relevant negative tests appropriate to the declared boundary and profile.

Testing SHOULD include, where applicable:

- forged Authority-State;
- tampered evidence;
- substituted Rule-State;
- expired governing state;
- revoked authority;
- execution-gate bypass;

- stale eligibility;
- duplicate execution;
- canonicalization mismatch;
- integrity-value mismatch;
- hidden-state dependency;
- external-dependency failure; and
- privileged override misuse.

**16.45 Adversarial Testing**

Profile II and Profile III implementations SHOULD be tested against attempts to produce consequence without valid execution eligibility.

Profile III implementations SHOULD additionally be tested against attempts to:

- suppress material change;
- suppress revocation;
- alter historical state;
- prevent accurate replay;
- introduce recomputation mismatch; and
- substitute current state for historical state.

**16.46 Security Evidence**

A conforming implementation SHALL preserve security evidence sufficient to demonstrate the security requirements applicable to its declared Conformance Profile.

Evidence MAY include:

- threat model;
- security-boundary documentation;
- access-control records;
- integrity tests;
- enforcement-bypass tests;

- revocation tests;
- replay-prevention tests;
- key-management evidence;
- incident records;
- remediation records; and
- security-assessment results.

**16.47 Vulnerability Handling**

A conforming implementation SHOULD maintain a governed process for identifying and remediating vulnerabilities capable of affecting Trust-State requirements.

The process SHOULD address:

- vulnerability intake;
- severity assessment;
- affected conformance boundary;
- remediation;
- verification;
- material-change determination;
- certification impact; and
- disclosure where applicable.

**16.48 Security Change Control**

Security-relevant changes capable of affecting conformance SHALL be governed under change control.

Such changes MAY include:

- key rotation;
- algorithm change;
- access-control change;
- enforcement change;
- dependency change;

- canonicalization change;
- logging change;
- revocation-source change; and
- privileged-role change.

**16.49 Security and Trust Separation**

Security SHALL NOT be treated as equivalent to Trust-State.

A secure system MAY still lack sufficient:

- authority;
- evidence;
- Rule-State;
- context;
- admissibility;
- continuity; or
- independent reproducibility.

Security protects the mechanisms through which governed determinations are made.

It does not itself establish admissibility.

**16.50 Conformance Requirement**

A system conforms to this section when it can demonstrate that security controls within its declared boundary protect the material governed conditions required by the applicable Conformance Profile against unauthorized creation, modification, substitution, suppression, bypass, rollback, replay, or loss to the degree necessary to preserve deterministic governance.

**16.51 Canonical Statement**

**Security protects the integrity of the governance path.**

An attacker SHALL NOT be able to create admissibility by forging authority, manufacturing evidence, altering rules, suppressing revocation, bypassing enforcement, or corrupting replay.

A Trust-State system is secure when the path from governing condition to consequence cannot be materially altered without detection or governed response.

## Section 17 — Lifecycle and Change Control

### 17.0 Purpose

This section defines the normative lifecycle and change-control requirements applicable to Trust-State governed implementations, specifications, rules, dependencies, conformance claims, and certification-relevant states.

Lifecycle governance ensures that material change does not silently invalidate previously established admissibility, execution eligibility, continuity, recomputation, conformance, or certification assumptions.

This section is normative.

### 17.1 Lifecycle Principle

A conforming implementation SHALL govern material change across the lifecycle of Trust-State governed objects and system components.

Lifecycle governance SHALL ensure that material changes are:

- identified;
- attributable;
- version-controlled where applicable;
- evaluated for impact;
- tested where required;
- approved where required;
- implemented in a controlled manner;
- reflected in affected conformance or certification records; and
- preserved with sufficient provenance.

### 17.2 Lifecycle Scope

Lifecycle governance SHALL apply, where material, to:

- Authority-State;
- evidence;
- Rule-State;
- admissibility logic;

- Trust-State derivation;
- execution eligibility;
- enforcement mechanisms;
- continuity logic;
- revocation handling;
- canonicalization procedures;
- integrity mechanisms;
- recomputation procedures;
- replay procedures;
- implementation components;
- external dependencies;
- conformance boundaries;
- conformance claims; and
- certification records.

**17.3 Change Classification**

A conforming implementation SHALL distinguish material change from non-material change.

Material change is a change capable of affecting:

- authority sufficiency;
- evidence sufficiency;
- Rule-State applicability;
- admissibility;
- Trust-State;
- execution eligibility;
- enforcement behavior;
- continuity;
- revocation behavior;
- canonical representation;

- integrity value;
- replay;
- recomputation;
- conformance; or
- certification.

**17.4 Non-Material Change**

A change MAY be classified as non-material where it cannot affect a governed determination or conformity result within the declared boundary.

Non-material changes MAY include:

- presentation-only changes;
- editorial corrections;
- documentation formatting;
- non-governing interface changes; and
- implementation changes demonstrated not to affect governed behavior.

The basis for non-material classification SHOULD be documented.

**17.5 Change Identity**

A material change SHALL be identifiable.

The change record SHALL include or reference, where applicable:

- change identifier;
- affected object or component;
- prior version;
- new version;
- change description;
- change authority;
- approval state;
- effective time;

- affected boundary; and
- impact determination.

**17.6 Version Control**

Governed objects and procedures SHALL be version-controlled where change can materially affect output or verification.

Version-controlled objects MAY include:

- Authority-State schemas;
- evidence schemas;
- Rule-State;
- evaluation procedures;
- canonicalization specifications;
- integrity procedures;
- enforcement logic;
- continuity logic;
- replay logic; and
- conformance test specifications.

**17.7 Version Identity**

A version identifier SHALL distinguish materially different governed states or procedures.

Version identity SHALL be sufficiently stable to support:

- independent verification;
- historical replay;
- change impact assessment;
- conformance assessment;
- certification review; and
- provenance.

### 17.8 Immutable Historical State

Historical governed state required for verification, recomputation, or replay SHALL NOT be silently overwritten by a successor state.

A successor version SHALL remain distinguishable from the version it replaces.

Historical preservation MAY use:

- immutable storage;
- append-only records;
- content-addressed storage;
- versioned repositories;
- signed records; or
- another mechanism sufficient to preserve historical identity and integrity.

### 17.9 Change Authority

A material change SHALL be performed or approved by an actor or mechanism possessing sufficient authority.

Technical ability to modify a governed object SHALL NOT by itself establish authority to make the change.

The authority basis for material change SHALL be identifiable where required.

### 17.10 Change Provenance

Material change provenance SHALL preserve sufficient information to determine:

- what changed;
- who or what initiated the change;
- who or what authorized the change;
- when the change became effective;
- why the change occurred where required;
- what boundary was affected; and
- what resulting state was produced.

**17.11 Change Impact Assessment**

A material change SHALL be evaluated for impact before or upon activation where required by the applicable Rule-State, Conformance Profile, or certification program.

Impact assessment SHALL consider, where applicable:

- Authority-State validity;
- evidence validity;
- Rule-State applicability;
- admissibility logic;
- Trust-State continuity;
- execution eligibility;
- enforcement behavior;
- revocation handling;
- canonicalization;
- integrity values;
- recomputation;
- replay;
- conformance status; and
- certification status.

**17.12 Change Before Activation**

Where a material change can affect consequence-bearing execution, the implementation SHOULD evaluate the change before activating it within the governed execution path.

Where pre-activation evaluation is required, the change SHALL NOT become active until applicable requirements are satisfied.

**17.13 Emergency Change**

A Rule-State MAY permit emergency change.

An emergency change process SHALL define:

- triggering conditions;

- authorized actors;
- permitted scope;
- required evidence;
- duration;
- compensating controls;
- retrospective review;
- normalization or rollback requirements; and
- provenance requirements.

Emergency status SHALL NOT eliminate change traceability.

**17.14 Change Testing**

Material changes SHALL be tested to the degree necessary to determine whether applicable Trust-State requirements remain satisfied.

Testing MAY include:

- positive testing;
- negative testing;
- regression testing;
- enforcement testing;
- continuity testing;
- revocation testing;
- canonicalization testing;
- integrity testing;
- recomputation testing; and
- replay testing.

### 17.15 Regression Testing

Where a change affects governed evaluation logic, regression testing SHALL demonstrate that previously valid behavior remains valid where intended and that previously prohibited behavior remains prohibited where intended.

Regression testing SHALL include applicable negative cases.

### 17.16 Rule-State Change Control

Material Rule-State changes SHALL satisfy the change-control requirements of Section 7.

A Rule-State change SHALL identify, where applicable:

- prior Rule-State;
- successor Rule-State;
- changed rule identities;
- effective time;
- transition behavior;
- applicability;
- supersession relationship;
- rollback conditions; and
- provenance.

### 17.17 Authority-State Change Control

Material Authority-State changes SHALL preserve:

- prior state identity;
- successor state identity;
- changed authority condition;
- change authority;
- effective time;
- revocation or supersession state;
- lineage; and

- provenance.

Authority-State change SHALL trigger continuity evaluation where required by Section 10.

**17.18 Evidence Change Control**

Evidence changes capable of affecting admissibility SHALL be governed.

Change MAY include:

- replacement;
- correction;
- expiration;
- revocation;
- supersession;
- provenance update;
- transformation; or
- integrity-state change.

The system SHALL preserve enough information to distinguish the evidence used historically from current evidence.

**17.19 Evaluation-Logic Change**

A change to admissibility or Trust-State evaluation logic SHALL be treated as material where it can alter governed output.

The changed logic SHALL be version-identifiable.

Historical replay SHALL use the historical procedure or a verified equivalent procedure.

**17.20 Enforcement Change**

A material change to execution enforcement SHALL be evaluated for whether it:

- changes the execution boundary;
- changes eligibility binding;
- creates a bypass;

- alters fail-closed behavior;
- alters expiry handling;
- alters revocation handling; or
- alters action-level enforcement.

A change that weakens mandatory enforcement SHALL constitute a material change.

**17.21 Continuity-Logic Change**

A change to continuity evaluation SHALL be material where it can alter:

- material-change detection;
- re-evaluation triggers;
- expiration behavior;
- suspension;
- revocation;
- revocation propagation;
- restoration; or
- stale-state handling.

**17.22 Canonicalization Change**

A canonicalization change SHALL satisfy Section 11.

A change capable of altering canonical serialized output SHALL produce a distinguishable canonicalization version.

The change SHALL address impact on:

- historical integrity values;
- replay;
- recomputation;
- cross-implementation equivalence;
- conformance tests; and
- stored artifacts.

**17.23 Integrity-Procedure Change**

A change to an integrity algorithm or procedure SHALL identify:

- prior algorithm;
- successor algorithm;
- effective time;
- transition requirements;
- compatibility;
- verification requirements;
- historical verification support; and
- affected objects.

**17.24 Dependency Change**

A material external-dependency change SHALL be subject to impact assessment.

Dependency change MAY include:

- provider change;
- interface change;
- data-source change;
- version change;
- trust-anchor change;
- availability change;
- schema change;
- historical-state loss; or
- security posture change.

**17.25 Dependency Replacement**

Where an external dependency is replaced, the system SHALL determine whether the replacement provides equivalent governed behavior where equivalence is required.

Dependency replacement SHALL NOT be treated as non-material merely because the external interface remains unchanged.

**17.26 Configuration Change**

Configuration changes capable of altering governed behavior SHALL be treated as material.

Material configuration MAY include:

- Rule-State selection;
- thresholds;
- evidence requirements;
- authority constraints;
- enforcement mode;
- fail-open or fail-closed behavior;
- revocation source;
- canonicalization mode;
- cryptographic configuration; and
- replay settings.

**17.27 Infrastructure Change**

Infrastructure change MAY be non-material where governed behavior is demonstrably preserved.

Where infrastructure can affect deterministic behavior, timing, state persistence, security, or historical reproducibility, impact SHALL be assessed.

**17.28 Migration**

A system migration SHALL preserve governed state necessary for continued conformity.

Migration SHALL preserve or verifiably transform, where applicable:

- Authority-State;
- evidence;
- Rule-State;

- Trust-State;
- execution eligibility;
- continuity history;
- revocation history;
- canonical representations;
- integrity values;
- replay data; and
- conformance evidence.

**17.29 State Transformation During Migration**

Where governed state is transformed during migration, the transformation SHALL be:

- defined;
- deterministic where required;
- version-identifiable;
- integrity-verifiable;
- traceable to prior state; and
- tested for material equivalence.

**17.30 System Replacement**

Replacement of a conforming system SHALL NOT automatically transfer conformity to the successor system.

The successor SHALL undergo impact assessment and conformance evaluation appropriate to the degree of change.

**17.31 Failover and Recovery**

Where failover or disaster recovery can alter the governed execution path, the implementation SHALL preserve applicable Trust-State requirements during failover.

Failover SHALL NOT introduce an ungoverned execution path.

**17.32 Recovery State**

Following recovery, the system SHALL determine whether:

- Authority-State remains current;
- evidence remains valid;
- Rule-State remains current;
- revocation state is current;
- Trust-State remains valid;
- eligibility remains valid; and
- continuity re-evaluation is required.

**17.33 Backup and Restore**

Where backup and restore are used, the system SHALL prevent restoration of stale governed state from silently creating current admissibility or eligibility.

Restored state SHALL be evaluated against applicable continuity and revocation requirements.

**17.34 Rollback**

Authorized rollback of implementation state SHALL be governed.

Rollback SHALL identify:

- rollback target;
- authorization;
- affected boundary;
- effect on Rule-State;
- effect on stored governing state;
- effect on conformance;
- effect on certification; and
- effective time.

Rollback SHALL NOT silently reactivate revoked, expired, or superseded governing state.

### 17.35 Decommissioning

A conforming implementation SHALL govern decommissioning where historical verification, replay, certification, or evidence-retention obligations remain.

Decommissioning SHALL address, where applicable:

- preservation of historical state;
- preservation of verification capability;
- retention periods;
- cryptographic verification;
- certification record status;
- archival access; and
- destruction after retention requirements expire.

### 17.36 Historical Verification After Decommissioning

Where historical verification remains required after system decommissioning, sufficient artifacts, specifications, keys, trust anchors, or other verification resources SHALL remain available for the declared retention period.

### 17.37 Conformance Impact

A material change SHALL be evaluated for its effect upon an existing conformance claim.

The resulting status MAY be:

- unaffected;
- reassessment required;
- conditionally conforming;
- suspended;
- non-conforming; or
- replaced by a new claim.

**17.38 Certification Impact**

A material change to a certified implementation SHALL be governed according to Section 15.

Certification impact MAY include:

- no action;
- notification;
- targeted reassessment;
- partial reassessment;
- full reassessment;
- suspension;
- renewal; or
- new certification.

**17.39 Conformance Boundary Change**

Expansion of a conformance boundary SHALL require evaluation of the newly included scope.

A conformance claim SHALL NOT automatically extend to:

- new action classes;
- new actors;
- new resources;
- new jurisdictions;
- new Rule-State domains;
- new dependencies; or
- new execution pathways.

**17.40 Profile Change**

A change affecting capability required by the declared Conformance Profile SHALL trigger profile impact assessment.

Loss of a required capability SHALL require correction of the conformance claim if conformity can no longer be demonstrated.

**17.41 Standard Version Change**

Publication of a successor Trust-State Standard release SHALL NOT automatically invalidate conformance to an earlier release.

A conformance claim SHALL continue to identify the specific release assessed.

Migration to a successor Standard release SHALL be governed as a new or updated conformity determination.

**17.42 Standard Supersession**

A successor Standard release MAY supersede an earlier release for future conformance or certification purposes.

Supersession rules SHALL define, where applicable:

- predecessor release;
- successor release;
- effective date;
- transition period;
- certification implications;
- claim implications; and
- continued historical status.

Supersession SHALL NOT alter the historical content of the predecessor release.

**17.43 Standard Immutability**

A published, version-identified Trust-State Standard release designated as immutable or final SHALL NOT be silently modified.

Corrections or substantive changes SHALL be issued through:

- errata where permitted;
- corrigenda where permitted;
- amendment;
- point release;
- successor release; or

- another formally identified publication mechanism.

Historical releases SHALL remain distinguishable from successor releases.

**17.44 Normative Annex Change**

A change to a normative annex SHALL be governed according to its relationship with the Standard release.

Where an annex change can alter conformance requirements, the applicable:

- annex version;
- Standard compatibility;
- transition rule;
- conformance effect; and
- certification effect

SHALL be identified.

**17.45 Informative Material Change**

Informative guidance MAY be revised independently where the revision does not alter normative requirements.

Informative material SHALL NOT be used to silently introduce new mandatory implementation requirements.

**17.46 Change Approval**

The applicable governance process SHALL define approval authority for material changes.

Approval requirements MAY vary by:

- object type;
- risk;
- affected profile;
- affected action class;
- certification status; and
- deployment environment.

**17.47 Segregation of Change Functions**

Where required by Rule-State or governance policy, the implementation SHALL separate:

- change proposal;
- technical implementation;
- testing;
- approval;
- deployment; and
- conformity review.

**17.48 Unauthorized Change**

Unauthorized material change SHALL be treated as a governance and security failure.

The system SHALL determine whether affected:

- Authority-State;
- evidence;
- Rule-State;
- Trust-State;
- execution eligibility;
- historical verification;
- conformance; or
- certification

remain valid.

**17.49 Change Failure**

A change failure occurs where change control does not preserve required Trust-State behavior.

Failure MAY include:

- undocumented material change;
- unversioned evaluation logic;
- silent Rule-State replacement;

- stale authority persistence;
- stale evidence persistence;
- enforcement bypass;
- canonicalization divergence;
- integrity mismatch;
- replay failure;
- conformance overstatement; or
- certification overstatement.

**17.50 Change Failure Response**

Material change failure SHALL trigger a governed response.

The response MAY include:

- rollback;
- suspension;
- re-evaluation;
- remediation;
- reassessment;
- conformance claim update;
- certification notification;
- certification suspension; or
- another defined response.

**17.51 Change Records**

A conforming implementation SHALL preserve change records sufficient to establish lifecycle provenance.

Records SHOULD include:

- change identifier;
- affected object;

- prior version;
- successor version;
- authority;
- approval;
- test evidence;
- impact assessment;
- deployment state;
- conformance effect;
- certification effect; and
- effective time.

**17.52 Change Record Integrity**

Material change records SHALL be integrity-protected where alteration could impair:

- independent verification;
- replay;
- conformance assessment;
- certification review; or
- historical reconstruction.

**17.53 Lifecycle Replay**

Where Profile III applies, lifecycle records SHALL support reconstruction of the material governed state applicable at a prior determination point.

A verifier SHALL be able to distinguish:

- state existing before a change;
- change event;
- state existing after the change; and
- effective time of the transition.

### 17.54 Lifecycle Recomputation

Where independent recomputation is required, change history SHALL preserve enough information to determine which governed versions applied to the recomputed determination.

### 17.55 Retention

Lifecycle and change-control evidence SHALL be retained according to the applicable:

- Rule-State;
- Conformance Profile;
- certification program;
- legal requirement;
- contractual requirement; or
- organizational retention policy.

The declared retention boundary SHALL not imply availability beyond the period actually supported.

### 17.56 Governance of the Trust-State Standard

The Trust-State Standard SHALL be maintained through a version-controlled publication and governance process.

The process SHALL preserve:

- release identity;
- normative and informative separation;
- terminology stability;
- profile continuity;
- annex status;
- change history;
- supersession relationships; and
- historical release integrity.

**17.57 Normative Change Discipline**

A normative change SHALL be identifiable as such.

Normative requirements SHALL NOT be materially altered through:

- explanatory text;
- informal guidance;
- web-page copy;
- marketing material;
- certification guidance;
- implementation examples; or
- informative annexes.

**17.58 Interpretation Governance**

Where interpretation of normative text is required, an interpretation SHALL NOT silently amend the Standard.

Formal interpretations SHOULD:

- identify the affected provision;
- explain the interpretation;
- preserve the original text;
- identify governance authority;
- identify applicability; and
- distinguish interpretation from normative amendment.

**17.59 Release History**

The publication process SHALL preserve release history sufficient to determine:

- publication sequence;
- release date;
- release status;
- supersession relationship;

- normative changes;
- corrigenda or amendments; and
- applicable annex versions.

**17.60 Conformance Requirement**

A system conforms to this section when material lifecycle changes within its declared boundary are:

- identifiable;
- authorized;
- version-controlled where required;
- impact-assessed;
- tested where required;
- traceable;
- integrity-protected where required;
- reflected in continuity, conformance, and certification state;
- reproducible for historical verification where required; and
- prevented from silently altering governed behavior.

**17.61 Canonical Statement**

**Deterministic governance requires deterministic change.**

A governed state is meaningful only if a verifier can determine what changed, when it changed, under whose authority it changed, and which version governed the action in question.

Change may supersede prior state. It SHALL NOT erase it.

**Annexes**

Annex A — Canonical Serialization Specification

Normative. Exact serialization, ordering, encoding, hashing, boundary rules, deterministic representation requirements.

Annex B — Conformance Test Vectors

Normative. Positive/negative vectors for Profiles I–III, expected outputs, integrity values, replay cases, fail-closed cases.

Annex C — Reference Algorithms and Evaluation Procedures

Normative. Rule evaluation order, admissibility derivation, eligibility derivation, continuity/revocation handling, recomputation procedures.

Annex D — Interoperability and Machine-Readable Artifacts

Normative. Artifact structures, identifiers, claim records, certification records, cross-system exchange, binding rules.

Annex E — Certification and Assessment Procedures

Normative for the certification program, but separate from implementation conformance. Assessor procedures, evidence packages, decision lifecycle, suspension/renewal/revocation.

Annex F — Consequence Science and External Framework Mapping

Informative. Layer 0/Consequence Science context, conceptual lineage, and later mappings to things like EU AI Act, NIST, OMB, etc. No new implementation requirements.

**Annex A — Canonical Serialization Specification**

**Normative**

**A.1 Purpose**

This Annex defines the canonical serialization requirements used to produce deterministic representations of Trust-State governed objects.

Canonical serialization supports:

- deterministic comparison;
- integrity-value generation;
- independent verification;
- recomputation;
- replay; and
- cross-implementation equivalence.

Where this Annex is applicable under the declared Conformance Profile, its requirements are normative.

**A.2 Canonicalization Principle**

A governed object subject to canonical serialization SHALL produce one deterministic serialized representation for materially equivalent governed content.

Equivalent governed inputs SHALL NOT produce different canonical representations solely because of:

- implementation language;
- platform;
- storage format;
- presentation format;
- field insertion order;
- whitespace;
- locale;
- timezone representation;
- numeric formatting; or
- other implementation-dependent variation.

**A.3 Canonicalization Boundary**

Each canonicalized object SHALL have a defined canonicalization boundary.

The boundary SHALL identify:

- included fields;
- required fields;
- optional fields;
- excluded fields;
- derived fields where applicable;
- referenced external objects; and
- non-governing presentation data.

Information material to a governed determination SHALL NOT be excluded where exclusion would prevent independent verification or recomputation.

**A.4 Character Encoding**

Canonical serialized content SHALL use UTF-8 encoding unless another encoding is expressly defined by an applicable normative specification.

Byte Order Marks SHALL NOT be included unless expressly required by that specification.

**A.5 Field Naming**

Canonical field names SHALL be stable and case-sensitive.

Aliases, display labels, localized names, or implementation-specific names SHALL be resolved to the applicable canonical field name before serialization.

**A.6 Field Ordering**

Object fields SHALL be serialized according to a deterministic ordering rule.

Unless otherwise defined by an applicable object specification, fields SHALL be ordered lexicographically by canonical field name using their encoded Unicode scalar values.

Field insertion order SHALL NOT determine canonical output.

**A.7 Collection Ordering**

Where collection order is semantically significant, the original governed order SHALL be preserved.

Where collection order is not semantically significant, the collection SHALL be normalized using a deterministic ordering rule defined for that collection.

An unordered set SHALL NOT produce different canonical representations solely because its implementation-specific storage order differs.

**A.8 Whitespace**

Insignificant whitespace SHALL NOT be included in the canonical serialized representation.

Whitespace occurring within governed string values SHALL remain part of the governed value unless normalization is expressly defined.

**A.9 String Representation**

Strings SHALL be serialized using their canonical encoded value.

Where Unicode normalization is required, the applicable normalization form SHALL be identified by the serialization specification.

A conforming implementation SHALL NOT silently apply implementation-specific normalization that can alter governed meaning.

**A.10 Boolean and Null Values**

Boolean values SHALL have one canonical representation for true and one canonical representation for false.

Null SHALL be distinguishable from:

- absent value;
- empty string;
- zero;
- false; and
- empty collection.

The applicable object schema SHALL determine whether null and absent values are semantically distinct.

**A.11 Numeric Representation**

Equivalent numeric values SHALL produce equivalent canonical representations.

Canonical numeric representation SHALL prohibit ambiguity resulting from:

- leading zeros;
- unnecessary positive signs;
- alternate exponent notation;
- locale-specific separators; or
- implementation-specific formatting.

Where precision or scale is governance-relevant, the applicable schema SHALL define it.

**A.12 Temporal Representation**

Governance-relevant timestamps SHALL use a deterministic temporal representation.

Unless otherwise defined, timestamps SHALL:

- identify UTC or an explicit offset;
- use an unambiguous calendar representation;
- define required precision; and
- avoid locale-dependent formatting.

Temporal normalization SHALL preserve the instant or interval represented.

### A.13 Identifier Representation

Governed identifiers SHALL be represented in a stable canonical form.

Where an identifier has multiple textual representations, the applicable specification SHALL define the canonical form.

Identifiers SHALL remain sufficient to distinguish materially different governed objects.

### A.14 Binary Content

Binary content included directly in a canonical object SHALL use a deterministic encoding defined by the applicable specification.

Where binary content is referenced rather than embedded, the representation SHALL include sufficient identity and integrity information to identify the referenced object.

### A.15 External References

An external reference used within a canonical representation SHALL identify the referenced object without material ambiguity.

Where integrity of the referenced object is required, the reference SHALL include or resolve to an integrity value.

A reference SHOULD include, where applicable:

- object identifier;
- version;
- object type;
- integrity value; and
- location or resolution mechanism.

### A.16 Authority-State Representation

A canonical Authority-State representation SHALL include or reference the material elements required by Section 5, including as applicable:

- Authority-State identifier;
- authority source;
- subject;
- action or action class;
- resource or scope;
- constraints;
- delegation lineage;
- temporal validity;
- revocation state;
- provenance; and
- version identity.

Materially different Authority-States SHALL produce distinguishable canonical representations.

**A.17 Evidence Representation**

Canonical evidence representation SHALL identify the exact evidence relied upon by a governed determination.

Each material evidence item SHALL include or reference, where applicable:

- evidence identifier;
- evidence type;
- source;
- provenance;
- temporal state;
- integrity value;
- transformation lineage;
- revocation state; and
- version identity.

**A.18 Evidence-Set Representation**

Where multiple evidence items are evaluated together, the canonical evidence-set representation SHALL identify the complete governed evidence set.

The evidence-set representation SHALL define deterministic ordering where evidence-item order is not semantically significant.

Addition, removal, replacement, or material modification of an evidence item SHALL produce a distinguishable evidence-set representation.

**A.19 Rule-State Representation**

A canonical Rule-State representation SHALL include or reference the material Rule-State elements required by Section 7.

These SHALL include, where applicable:

- Rule-State identifier;
- rule identities;
- policy identities;
- versions;
- applicability;
- constraints;
- precedence;
- conflict-resolution rules;
- effective time;
- supersession state;
- provenance; and
- integrity information.

A Rule-State bound to a completed determination SHALL remain historically reproducible.

**A.20 Context Representation**

Context SHALL be canonicalized where context materially affects a governed determination.

Only governance-relevant context need be represented.

Material context MAY include:

- time;
- jurisdiction;
- actor state;
- resource state;
- transaction state;
- dependency state;

- environmental state; and
- execution state.

**A.21 Admissibility Determination Representation**

A canonical admissibility determination SHALL identify or reference:

- determination identifier;
- action identifier;
- Authority-State identifier;
- evidence-set identifier;
- Rule-State identifier;
- relevant context;
- evaluation-procedure version;
- determination outcome;
- Trust-State result where applicable;
- determination time; and
- applicable integrity values.

**A.22 Execution Eligibility Representation**

Where execution eligibility is represented canonically, the representation SHALL identify or reference:

- eligibility identifier;
- admissibility determination;
- governed action;
- subject;
- execution scope;
- applicable Trust-State;
- validity period;
- revocation state;
- conditions;
- one-time or reusable status where applicable; and
- resulting eligibility state.

**A.23 Continuity and Revocation Representation**

Canonical continuity and revocation representations SHALL preserve sufficient information to establish:

- affected prior state;
- material change;
- revocation source where applicable;
- effective time;
- affected scope;
- applicable Rule-State;
- resulting state; and
- propagation effects where applicable.

**A.24 Integrity Value Generation**

Where an integrity value is required, it SHALL be derived from the complete canonical serialized representation within the applicable integrity boundary.

The integrity procedure SHALL identify:

- algorithm;
- algorithm version or identifier;
- canonicalization version;
- object type; and
- serialized input.

**A.25 Cryptographic Hash Function**

The applicable conformance specification SHALL identify the approved cryptographic hash function or functions.

An implementation SHALL NOT substitute a different algorithm while claiming equivalent integrity output unless the applicable specification expressly permits it.

**A.26 Domain Separation**

Where integrity values are generated for different governed object types using the same cryptographic function, the implementation SHOULD use deterministic domain separation.

Domain separation MAY include an object-type identifier or canonical namespace within the hashed representation.

### A.27 Canonicalization Version

Every canonical representation used for verification, integrity generation, recomputation, or replay SHALL be associated with an identifiable canonicalization specification version.

A change capable of altering canonical output SHALL produce a new canonicalization version.

### A.28 Historical Serialization

Historical replay SHALL use:

- the canonicalization procedure applicable to the original determination; or
- a formally defined compatibility procedure proven to preserve the required historical representation.

A successor serialization rule SHALL NOT silently replace the historical rule.

### A.29 Cross-Implementation Requirement

Independent conforming implementations using:

- equivalent governed inputs;
- the same canonicalization version; and
- the same applicable object specification

SHALL produce equivalent canonical serialized representations.

Where the same integrity procedure applies, they SHALL also produce equivalent integrity values.

### A.30 Serialization Failure

Where required canonical serialization cannot be completed, the affected object SHALL NOT be treated as having a valid canonical representation.

Where canonical representation is mandatory for a governed determination, unresolved serialization failure SHALL prevent the dependent deterministic claim.

**A.31 Verification**

An independent verifier SHALL be able to determine, where required:

- canonicalization version;
- canonicalization boundary;
- governed input;
- canonical serialized representation;
- integrity algorithm;
- integrity value; and
- verification result.

**A.32 Test Vectors**

Normative or certification test suites MAY provide canonical serialization test vectors.

A test vector SHOULD identify:

- source governed object;
- canonical serialized form;
- expected integrity value;
- canonicalization version; and
- expected success or failure state.

**A.33 Conformance**

An implementation conforms to this Annex where the canonical representations within its declared conformance boundary are:

- deterministically generated;
- version-identifiable;
- materially complete;
- reproducible;
- integrity-verifiable where required; and
- equivalent across conforming implementations when supplied equivalent governed inputs.

### A.34 Canonical Statement

Deterministic governance requires a representation that does not change merely because the implementation changes.

Canonical serialization provides the stable representation through which governed state can be identified, integrity-protected, recomputed, and verified independently.

## Annex B — Conformance Test Vectors

**Normative**

### B.1 Purpose

This Annex defines normative conformance test vectors for evaluating implementation conformity with the Trust-State Standard.

The test vectors in this Annex are designed to verify deterministic behavior across:

- Authority-State evaluation;
- evidence sufficiency;
- Rule-State selection;
- admissibility determination;
- Trust-State derivation;
- execution eligibility;
- enforcement;
- continuity;
- revocation;
- canonical serialization;
- integrity verification;
- independent recomputation; and
- replay.

Where this Annex is applicable to a declared Conformance Profile, the expected results defined herein are normative.

### B.2 Test Vector Principle

A conforming implementation SHALL produce the expected governed result for each applicable test vector.

A test vector SHALL define, where applicable:

- test identifier;
- applicable Conformance Profile;
- action;
- Authority-State;
- evidence;
- Rule-State;
- context;
- dependencies;
- revocation state;
- expected admissibility result;

- expected Trust-State;
- expected execution eligibility;
- expected enforcement result;
- expected canonicalization result;
- expected integrity result;
- expected continuity result;
- expected replay result; and
- expected PASS or FAIL condition.

**B.3 Test Determinism**

Repeated execution of an applicable test vector under equivalent governing conditions SHALL produce equivalent governed results.

Variation in:

- implementation language;
- platform;
- storage engine;
- process instance;
- evaluator instance;
- deployment environment; or
- execution time

SHALL NOT alter the expected result unless such variation is itself a governed input.

**B.4 Test Result Classification**

Each applicable test vector SHALL result in one of:

- PASS;
- FAIL;
- NOT_APPLICABLE; or
- INDETERMINATE_TEST

where the test harness cannot validly execute the vector.

A conforming implementation SHALL NOT report PASS where the expected governed outcome is not produced.

**B.5 Test Vector Structure**

Each normative test vector SHOULD identify:

- Vector ID;
- Profile;
- Purpose;
- Inputs;
- Governing Conditions;
- Expected Intermediate State;
- Expected Final State;
- Expected Enforcement Behavior where applicable;
- Expected Verification Result;
- Failure Condition.

**B.6 Profile Applicability**

Profile I implementations SHALL execute all applicable Profile I vectors.

Profile II implementations SHALL execute:

- all applicable Profile I vectors; and
- all applicable Profile II vectors.

Profile III implementations SHALL execute:

- all applicable Profile I vectors;
- all applicable Profile II vectors; and
- all applicable Profile III vectors.

**B.7 Profile I — Valid Admissibility**

**Vector ID:** B-P1-001

**Purpose:** Verify deterministic admissibility under sufficient governing conditions.

**Inputs:**

- valid action identity;
- valid Authority-State;
- sufficient evidence;
- valid Rule-State;
- valid context;

- satisfied dependencies;
- no applicable revocation.

**Expected Result:**

- authority = SUFFICIENT;
- evidence = SUFFICIENT;
- Rule-State = VALID;
- context = VALID;
- admissibility = ADMISSIBLE;
- Trust-State = VALID.

**PASS Condition:**

The implementation derives the expected admissibility and Trust-State.

**B.8 Profile I — Missing Authority**

**Vector ID:** B-P1-002

**Purpose:** Verify that identity or access does not substitute for authority.

**Inputs:**

- valid actor identity;
- valid technical access;
- no sufficient Authority-State;
- sufficient evidence;
- valid Rule-State;
- valid context.

**Expected Result:**

- authority = INSUFFICIENT;
- admissibility = NON_ADMISSIBLE or equivalent defined failure state;
- Trust-State SHALL NOT be VALID.

**PASS Condition:**

The implementation does not establish admissibility solely from identity or access.

**B.9 Profile I — Out-of-Scope Authority**

**Vector ID:** B-P1-003

**Purpose:** Verify action-specific authority evaluation.

**Inputs:**

- valid Authority-State for Action Class A;
- proposed Action Class B;
- otherwise sufficient evidence and Rule-State.

**Expected Result:**

- authority = INSUFFICIENT;
- admissibility = NON_ADMISSIBLE.

**B.10 Profile I — Expired Authority**

**Vector ID:** B-P1-004

**Inputs:**

- Authority-State valid until T1;
- action request occurs after T1.

**Expected Result:**

- Authority-State = EXPIRED;
- authority = INSUFFICIENT;
- admissibility = NON_ADMISSIBLE.

**B.11 Profile I — Revoked Authority**

**Vector ID:** B-P1-005

**Inputs:**

- otherwise valid Authority-State;
- revocation effective before evaluation.

**Expected Result:**

- authority = REVOKED or INSUFFICIENT;
- admissibility = NON_ADMISSIBLE.

**B.12 Profile I — Invalid Delegation Lineage**

**Vector ID:** B-P1-006

**Inputs:**

- delegated Authority-State;
- source authority does not permit required sub-delegation or delegated scope exceeds source scope.

**Expected Result:**

- authority lineage = INVALID;
- authority = INSUFFICIENT;
- admissibility = NON_ADMISSIBLE.

**B.13 Profile I — Valid Evidence**

**Vector ID:** B-P1-007

**Inputs:**

- required evidence items present;
- evidence valid;
- evidence applicable;
- provenance valid;
- integrity valid.

**Expected Result:**

- evidence = SUFFICIENT.

**B.14 Profile I — Missing Required Evidence**

**Vector ID:** B-P1-008

**Inputs:**

- valid Authority-State;
- one mandatory evidence item absent.

**Expected Result:**

- evidence = INSUFFICIENT;
- admissibility = NON_ADMISSIBLE or INDETERMINATE as defined by Rule-State;
- Trust-State SHALL NOT be VALID.

**B.15 Profile I — Invalid Evidence**

**Vector ID:** B-P1-009

**Inputs:**

- evidence item present;
- evidence integrity verification fails.

**Expected Result:**

- evidence = INVALID;
- evidence sufficiency = INSUFFICIENT;
- admissibility SHALL NOT be established.

**B.16 Profile I — Expired Evidence**

**Vector ID:** B-P1-010

**Inputs:**

- evidence otherwise valid;
- evidence validity period has expired before evaluation.

**Expected Result:**

- evidence = EXPIRED;
- evidence SHALL NOT satisfy a requirement requiring current evidence.

**B.17 Profile I — Revoked Evidence**

**Vector ID:** B-P1-011

**Inputs:**

- evidence originally valid;
- revocation effective before determination.

**Expected Result:**

- evidence = REVOKED;
- sufficiency SHALL be recomputed;
- admissibility SHALL fail where no sufficient alternative evidence exists.

**B.18 Profile I — Evidence Exists but Is Inapplicable**

**Vector ID:** B-P1-012

**Purpose:** Verify that existence does not imply applicability or sufficiency.

**Inputs:**

- evidence valid in form;
- evidence applies to a different actor, resource, action, time, or governing condition.

**Expected Result:**

- evidence = INAPPLICABLE or equivalent;
- evidence = INSUFFICIENT.

**B.19 Profile I — Invalid Rule-State**

**Vector ID:** B-P1-013

**Inputs:**

- authority sufficient;
- evidence sufficient;
- Rule-State integrity cannot be verified.

**Expected Result:**

- Rule-State = INVALID;
- admissibility SHALL NOT be established.

**B.20 Profile I — Superseded Rule-State Used Improperly**

**Vector ID:** B-P1-014

**Inputs:**

- Rule-State R1 superseded by R2 before current evaluation;
- system attempts current evaluation under R1 without governed authorization.

**Expected Result:**

- Rule-State selection = INVALID;
- current admissibility SHALL NOT be based on R1.

**B.21 Profile I — Rule Conflict with Defined Precedence**

**Vector ID:** B-P1-015

**Inputs:**

- two applicable rules conflict;
- Rule-State defines deterministic precedence.

**Expected Result:**

- conflict resolved using defined precedence;
- equivalent implementations produce equivalent result.

**B.22 Profile I — Rule Conflict Without Resolution**

**Vector ID:** B-P1-016

**Inputs:**

- conflicting applicable rules;
- no governing precedence or conflict-resolution mechanism.

**Expected Result:**

- Rule-State evaluation = INDETERMINATE;
- admissibility SHALL NOT be affirmatively established where deterministic admissibility is required.

**B.23 Profile I — Invalid Context**

**Vector ID:** B-P1-017

**Inputs:**

- authority and evidence sufficient;
- required contextual condition not satisfied.

**Expected Result:**

- context = INVALID;
- admissibility = NON_ADMISSIBLE.

**B.24 Profile I — Missing Required Context**

**Vector ID:** B-P1-018

**Inputs:**

- required contextual value cannot be established.

**Expected Result:**

- context = INDETERMINATE;
- admissibility SHALL NOT be established where context is mandatory.

**B.25 Profile I — Failed Dependency**

**Vector ID:** B-P1-019

**Inputs:**

- mandatory dependency state = UNSATISFIED.

**Expected Result:**

- dependencies = UNSATISFIED;
- admissibility = NON_ADMISSIBLE.

**B.26 Profile I — Probabilistic Input Within Bound**

**Vector ID:** B-P1-020

**Inputs:**

- probabilistic system output supplied as governed evidence input;
- Rule-State defines source, threshold, role, and interpretation.

**Expected Result:**

- probabilistic output is treated as a governed input;
- final admissibility remains deterministically derived under Rule-State.

**B.27 Profile I — Unbounded Probabilistic Input**

**Vector ID:** B-P1-021

**Inputs:**

- probabilistic score materially affects admissibility;
- Rule-State does not define threshold or deterministic interpretation.

**Expected Result:**

- admissibility = INDETERMINATE or NON_ADMISSIBLE;
- implementation SHALL NOT infer affirmative admissibility from the score alone.

**B.28 Profile I — Equivalent Input Determinism**

**Vector ID:** B-P1-022

**Inputs:**

Two semantically equivalent governed input sets represented using permitted implementation-level variations.

**Expected Result:**

Equivalent admissibility and Trust-State outputs.

**B.29 Profile II — Eligible Execution**

**Vector ID:** B-P2-001

**Inputs:**

- Profile I result = ADMISSIBLE;
- Trust-State valid;
- execution conditions satisfied;
- eligibility within validity window.

**Expected Result:**

- execution eligibility = ELIGIBLE;
- execution gate = PERMIT.

**B.30 Profile II — Non-Admissible Execution Attempt**

**Vector ID:** B-P2-002

**Inputs:**

- admissibility = NON_ADMISSIBLE;
- execution requested.

**Expected Result:**

- execution eligibility = NOT_ELIGIBLE;
- execution gate = DENY;
- consequence-bearing execution SHALL NOT occur through the governed path.

**B.31 Profile II — Indeterminate Admissibility**

**Vector ID:** B-P2-003

**Inputs:**

- admissibility = INDETERMINATE;
- established admissibility required for execution.

**Expected Result:**

- eligibility SHALL NOT be ELIGIBLE;
- enforcement SHALL deny execution.

**B.32 Profile II — Expired Eligibility**

**Vector ID:** B-P2-004

**Inputs:**

- eligibility valid until T1;
- execution request occurs after T1.

**Expected Result:**

- eligibility = EXPIRED;
- execution gate = DENY.

**B.33 Profile II — Revocation Before Execution**

**Vector ID:** B-P2-005

**Inputs:**

- action previously admissible and eligible;
- required authority revoked before execution.

**Expected Result:**

- prior eligibility invalidated;
- execution gate = DENY.

**B.34 Profile II — Material Change Before Execution**

**Vector ID:** B-P2-006

**Inputs:**

- valid eligibility derived;
- material governing condition changes before execution;
- Rule-State requires re-evaluation.

**Expected Result:**

- prior eligibility SHALL NOT be used;
- re-evaluation required;
- execution prevented until new eligibility exists.

**B.35 Profile II — Action Substitution**

**Vector ID:** B-P2-007

**Inputs:**

- valid eligibility for Action A;
- attempt to use eligibility for Action B.

**Expected Result:**

- action binding failure;
- execution gate = DENY.

**B.36 Profile II — Resource Substitution**

**Vector ID:** B-P2-008

**Inputs:**

- eligibility valid for Resource R1;
- attempt to execute equivalent action against Resource R2.

**Expected Result:**

- scope binding failure;
- execution gate = DENY unless Rule-State expressly permits the broader scope.

**B.37 Profile II — Actor Substitution**

**Vector ID:** B-P2-009

**Inputs:**

- eligibility derived for Subject S1;
- execution attempted by S2.

**Expected Result:**

- subject binding failure;
- execution gate = DENY.

**B.38 Profile II — Execution Gate Bypass**

**Vector ID:** B-P2-010

**Inputs:**

- attempt to invoke consequence-bearing backend path without passing required eligibility enforcement point.

**Expected Result:**

- bypass prevented or detected;
- execution SHALL NOT succeed through an ungoverned path.

**B.39 Profile II — Enforcement Dependency Failure**

**Vector ID:** B-P2-011

**Inputs:**

- required eligibility verification service unavailable;
- Rule-State requires established eligibility.

**Expected Result:**

- fail-closed behavior;
- execution gate = DENY.

**B.40 Profile II — Duplicate Execution**

**Vector ID:** B-P2-012

**Inputs:**

- single-use eligibility already consumed;
- identical execution attempted again.

**Expected Result:**

- duplicate execution denied.

**B.41 Profile II — Reusable Eligibility Within Scope**

**Vector ID:** B-P2-013

**Inputs:**

- Rule-State explicitly permits reusable eligibility;
- repeated action remains within scope, time, and other governing constraints.

**Expected Result:**

- execution MAY remain eligible according to Rule-State.

**B.42 Profile II — Reusable Eligibility Outside Scope**

**Vector ID:** B-P2-014

**Inputs:**

- reusable eligibility exists;
- subsequent request exceeds defined scope or constraints.

**Expected Result:**

- eligibility = NOT_ELIGIBLE;
- execution denied.

**B.43 Profile II — Governed Override**

**Vector ID:** B-P2-015

**Inputs:**

- normal eligibility absent;
- Rule-State defines an authorized override process;
- override authority, evidence, scope, and provenance valid.

**Expected Result:**

- execution permitted only according to governed override conditions;
- override event preserved.

**B.44 Profile II — Ungoverned Override**

**Vector ID:** B-P2-016

**Inputs:**

- privileged actor attempts to bypass enforcement without Rule-State-defined override authority.

**Expected Result:**

- override rejected;
- execution denied;
- security-relevant event generated where required.

**B.45 Profile III — Continuity Without Material Change**

**Vector ID:** B-P3-001

**Inputs:**

- prior Trust-State valid;
- no material governing condition changes.

**Expected Result:**

- continuity = CONTINUING;
- Trust-State remains valid within applicable validity conditions.

**B.46 Profile III — Material Authority Change**

**Vector ID:** B-P3-002

**Inputs:**

- prior Trust-State valid;
- Authority-State materially changes.

**Expected Result:**

- continuity evaluation triggered;
- re-evaluation performed where required;
- resulting Trust-State updated.

**B.47 Profile III — Material Evidence Change**

**Vector ID:** B-P3-003

**Inputs:**

- evidence supporting prior Trust-State becomes invalid or revoked.

**Expected Result:**

- evidence sufficiency re-evaluated;
- dependent Trust-State invalidated, degraded, suspended, or re-derived according to Rule-State.

**B.48 Profile III — Material Rule-State Change**

**Vector ID:** B-P3-004

**Inputs:**

- active Rule-State changes;
- change applies to existing active Trust-State.

**Expected Result:**

- continuity evaluation triggered;
- applicability and resulting state determined under transition rules.

**B.49 Profile III — Trust-State Expiration**

**Vector ID:** B-P3-005

**Inputs:**

- Trust-State validity ends at T1;
- continued reliance attempted after T1.

**Expected Result:**

- Trust-State = EXPIRED;
- prior Trust-State SHALL NOT support current eligibility.

**B.50 Profile III — Revocation Propagation**

**Vector ID:** B-P3-006

**Inputs:**

- source Authority-State revoked;
- dependent delegated Authority-State exists;
- dependent Trust-State exists.

**Expected Result:**

- revocation propagates according to Rule-State;
- affected dependent states re-evaluated or invalidated;
- stale execution eligibility prevented.

**B.51 Profile III — Revocation Does Not Erase History**

**Vector ID:** B-P3-007

**Inputs:**

- action validly executed at T1;
- governing authority revoked at T2 after execution;
- revocation has no retrospective effect.

**Expected Result:**

- historical determination at T1 remains historically valid;
- current or future reliance after T2 reflects revocation;
- historical record remains unchanged.

**B.52 Profile III — Historical Rule-State Replay**

**Vector ID:** B-P3-008

**Inputs:**

- determination originally performed under Rule-State R1;
- current Rule-State = R2;
- replay requested for original determination.

**Expected Result:**

- replay uses R1;
- R2 SHALL NOT replace R1;
- replay result equivalent to historical expected result.

**B.53 Profile III — Historical Evidence Replay**

**Vector ID:** B-P3-009

**Inputs:**

- evidence valid at T1;
- evidence expired at T2;
- replay of determination at T1.

**Expected Result:**

- replay evaluates evidence according to its T1 state;
- current expiration SHALL NOT alter historical replay result.

**B.54 Profile III — Historical Authority Replay**

**Vector ID:** B-P3-010

**Inputs:**

- authority valid at T1;
- authority revoked at T2;
- replay of T1 determination.

**Expected Result:**

- replay evaluates historical authority state at T1;
- current revocation does not silently rewrite historical authority.

**B.55 Profile III — Current Re-Evaluation After Revocation**

**Vector ID:** B-P3-011

**Inputs:**

- same action class as historical action;
- authority now revoked.

**Expected Result:**

- current re-evaluation uses current revocation state;
- current result MAY differ from historical replay;
- current admissibility SHALL fail where revoked authority was mandatory.

**B.56 Profile III — Replay Versus Re-Evaluation**

**Vector ID:** B-P3-012

**Purpose:** Verify separation of historical replay and current re-evaluation.

**Inputs:**

- same logical action;
- historical governing state differs materially from current governing state.

**Expected Result:**

- historical replay uses historical state;
- current re-evaluation uses current state;
- differing results SHALL NOT constitute mismatch when both are correct under their respective governing conditions.

**B.57 Profile III — Independent Recomputation Match**

**Vector ID:** B-P3-013

**Inputs:**

- preserved governed inputs;
- independent evaluator;
- equivalent evaluation logic.

**Expected Result:**

- recomputed admissibility equivalent to original;
- recomputed Trust-State equivalent to original;
- required integrity values equivalent where applicable.

**B.58 Profile III — Recomputation Input Mismatch**

**Vector ID:** B-P3-014

**Inputs:**

- independent evaluator receives materially different evidence set.

**Expected Result:**

- recomputation discrepancy identified;
- result SHALL NOT be represented as equivalent recomputation.

**B.59 Profile III — Hidden-State Dependency**

**Vector ID:** B-P3-015

**Inputs:**

- original determination depends upon undocumented internal state unavailable to independent evaluator.

**Expected Result:**

- independent recomputation cannot be demonstrated;
- test = FAIL for a profile requiring reproducibility.

**B.60 Profile III — Nondeterministic Governing Result**

**Vector ID:** B-P3-016

**Inputs:**

- equivalent governed inputs repeatedly evaluated;
- materially different admissibility results produced without governed input change.

**Expected Result:**

- NONDETERMINISTIC_RESULT;
- test = FAIL.

**B.61 Profile III — Preserved Probabilistic Output Replay**

**Vector ID:** B-P3-017

**Inputs:**

- historical determination used probabilistic score P1 as a governed input;
- underlying model now produces P2.

**Expected Result:**

- replay uses preserved P1 where P1 was the historical governed input;
- model SHALL NOT be required to regenerate P1 unless expressly required by Rule-State.

**B.62 Profile III — Human Decision Replay**

**Vector ID:** B-P3-018

**Inputs:**

- governed human approval formed part of historical input;
- provenance preserved.

**Expected Result:**

- replay uses the preserved governed approval;
- replay does not require reconstruction of undisclosed human reasoning unless that reasoning was itself a required governed input.

**B.63 Canonical Serialization — Equivalent Object**

**Vector ID:** B-CS-001

**Purpose:** Verify cross-implementation canonical equivalence.

**Inputs:**

Two materially equivalent objects with:

- different field insertion order;
- different insignificant whitespace;
- equivalent numeric formatting;
- equivalent allowed temporal formatting prior to canonicalization.

**Expected Result:**

- identical canonical representation under Annex A;
- identical integrity value where same integrity procedure applies.

**B.64 Canonical Serialization — Material Difference**

**Vector ID:** B-CS-002

**Inputs:**

Two objects differing in a material governed field.

**Expected Result:**

- distinguishable canonical representations;
- distinguishable integrity values where cryptographic hashing applies.

**B.65 Canonical Serialization — Missing Material Field**

**Vector ID:** B-CS-003

**Inputs:**

- canonical object missing a required governance-relevant field.

**Expected Result:**

- serialization or schema validation failure;
- dependent deterministic claim SHALL NOT be treated as valid.

**B.66 Canonical Serialization — Unknown Presentation Field**

**Vector ID:** B-CS-004

**Inputs:**

- governed object contains additional field expressly outside canonicalization boundary.

**Expected Result:**

- canonical representation remains equivalent if the field is non-governing and properly excluded.

**B.67 Canonical Serialization — Cross-Runtime Numeric Drift**
**Vector ID: B-CS-005**
Purpose: Verify deterministic numeric representation across independent runtime environments.

Inputs:
Two conforming implementations receive semantically equivalent governed numeric values represented through materially different native runtime forms, including where applicable:

- integer versus decimal representation;
- trailing-zero variation;
- scientific notation;
- platform-specific floating-point representation; or
- equivalent scale representation permitted by the applicable schema.

Expected Result:

- equivalent canonical numeric representation under the applicable canonicalization specification;
- equivalent canonical object representation;
- equivalent integrity value where the same integrity procedure applies;
- test = FAIL where platform-specific numeric behavior alters the governed canonical result.

**B.68 Canonical Serialization — Unknown Governing Attribute**
**Vector ID:** B-CS-006
Purpose: Verify deterministic handling of an unrecognized attribute that cannot be established as non-governing.

Inputs:

- governed object contains an attribute not recognized by the applicable object schema;
- the attribute is not expressly excluded from the canonicalization boundary;
- the implementation cannot establish that the attribute is semantically irrelevant.

Expected Result:

- implementation applies the defined unknown-attribute rule;
- the attribute SHALL NOT be silently discarded;
- where deterministic treatment cannot be established, serialization = FAIL or INDETERMINATE according to the applicable specification;
- dependent deterministic claim SHALL NOT be treated as valid where canonical representation is required.

**B.69 Rule-State — Unresolved Cyclic Dependency**
**Vector ID:** B-RS-001
Purpose: Verify deterministic handling of cyclic rule or policy dependencies.

Inputs:

- Rule-State R1 references R2;
- R2 directly or transitively references R1;
- the applicable Rule-State or evaluation procedure defines no deterministic terminating resolution.

Expected Result:

- cyclic dependency detected;
- Rule-State condition = INDETERMINATE or insufficient according to the applicable procedure;
- admissibility SHALL NOT be asserted where Rule-State sufficiency is required;
- implementation SHALL NOT resolve the cycle through undocumented recursion, lookup order, caching, or implementation-specific fallback behavior.

Then the existing integrity tests move down accordingly.

**B.67 Integrity — Valid Representation**

**Vector ID:** B-INT-001

**Inputs:**

- canonical object;
- expected integrity algorithm;
- expected integrity value.

**Expected Result:**

- integrity = VALID.

**B.68 Integrity — Tampered Representation**

**Vector ID:** B-INT-002

**Inputs:**

- canonical object modified after integrity value generated.

**Expected Result:**

- integrity verification = FAIL;
- affected governed object SHALL NOT be relied upon where integrity is mandatory.

**B.69 Integrity — Algorithm Mismatch**

**Vector ID:** B-INT-003

**Inputs:**

- claimed integrity value generated using algorithm different from declared applicable algorithm.

**Expected Result:**

- integrity verification = FAIL.

**B.70 Security — Forged Authority-State**

**Vector ID:** B-SEC-001

**Inputs:**

- Authority-State content structurally valid;
- source authenticity or integrity invalid.

**Expected Result:**

- Authority-State SHALL NOT establish authority sufficiency.

**B.71 Security — Rule-State Substitution**

**Vector ID:** B-SEC-002

**Inputs:**

- attacker substitutes less restrictive Rule-State.

**Expected Result:**

- substitution detected or rejected;
- improper admissibility SHALL NOT result.

**B.72 Security — Revocation Suppression**

**Vector ID:** B-SEC-003

**Inputs:**

- authority revoked;
- revocation information intentionally unavailable or suppressed;
- current revocation status mandatory.

**Expected Result:**

- current authority SHALL NOT be presumed valid;
- fail-closed behavior where required.

**B.73 Security — Stale Eligibility Replay**

**Vector ID:** B-SEC-004

**Inputs:**

- previously valid execution eligibility;
- eligibility expired, consumed, revoked, or invalidated;
- reuse attempted.

**Expected Result:**

- execution denied.

**B.74 Security — Unauthorized Rule Rollback**

**Vector ID:** B-SEC-005

**Inputs:**

- current Rule-State R2;
- unauthorized attempt to reactivate R1.

**Expected Result:**

- rollback rejected;
- governing Rule-State remains R2.

**B.75 Security — Cross-Boundary Artifact Reuse**

**Vector ID:** B-SEC-006

**Inputs:**

- valid eligibility artifact for Tenant A;
- attempted use in Tenant B.

**Expected Result:**

- boundary mismatch detected;
- execution denied.

**B.76 Conformance Claim — Valid Claim**

**Vector ID:** B-CLM-001

**Inputs:**

Conformance claim identifies:

- TSS version;
- Profile;
- implementation;
- boundary;
- evidence;
- status.

**Expected Result:**

- claim structure = VALID.

**B.77 Conformance Claim — Unqualified Claim**

**Vector ID:** B-CLM-002

**Inputs:**

Claim states only:

“Trust-State compliant.”

**Expected Result:**

- claim = NON_CONFORMING under Section 14;
- version, profile, and boundary missing.

**B.78 Conformance Claim — Profile Overstatement**

**Vector ID:** B-CLM-003

**Inputs:**

- implementation demonstrates Profile I only;
- claim states Profile II.

**Expected Result:**

- claim = FALSE_OR_MISLEADING;
- conformance result = FAIL.

**B.79 Certification — Active Valid Record**

**Vector ID:** B-CERT-001

**Inputs:**

Certification record identifies:

- implementation;
- Standard version;
- Profile;
- boundary;
- certification body;
- certification identifier;
- ACTIVE status.

**Expected Result:**

- certification record valid if supporting evidence and integrity verify.

**B.80 Certification — Suspended Mark Use**

**Vector ID:** B-CERT-002

**Inputs:**

- certification status = SUSPENDED;
- certification mark remains publicly used as active.

**Expected Result:**

- certification-program violation;
- test = FAIL.

**B.81 Certification — Boundary Overstatement**

**Vector ID:** B-CERT-003

**Inputs:**

- certified boundary = Action Class A;
- certification claim applied publicly to Action Classes A, B, and C.

**Expected Result:**

- certification claim exceeds certified boundary;
- test = FAIL.

**B.82 Lifecycle — Material Change Properly Governed**

**Vector ID:** B-LC-001

**Inputs:**

- evaluation logic materially changed;
- new version created;
- impact assessment completed;
- regression testing completed;
- conformance state updated.

**Expected Result:**

- lifecycle governance = PASS.

**B.83 Lifecycle — Silent Evaluation Logic Change**

**Vector ID:** B-LC-002

**Inputs:**

- evaluation logic materially altered;
- no version change;
- historical procedure overwritten.

**Expected Result:**

- lifecycle governance = FAIL;
- historical replay integrity impaired.

**B.84 Lifecycle — Historical State Preserved**

**Vector ID:** B-LC-003

**Inputs:**

- Rule-State R1 replaced by R2;

- historical determination under R1 preserved.

**Expected Result:**

- R1 remains reproducibly identifiable;
- replay under R1 remains possible within retention boundary.

**B.85 Lifecycle — Stale Backup Restore**

**Vector ID:** B-LC-004

**Inputs:**

- system restored from backup containing previously valid but now revoked authority and stale Rule-State.

**Expected Result:**

- restored state SHALL NOT automatically become current;
- continuity/revocation validation performed before reliance;
- stale eligibility SHALL NOT become executable.

**B.86 Cross-Profile Negative Test**

**Vector ID:** B-XP-001

**Purpose:** Verify that higher-profile claims include lower-profile capability.

**Inputs:**

- implementation claims Profile III;
- fails a mandatory Profile I authority test.

**Expected Result:**

- Profile III conformance = FAIL.

**B.87 Cross-Profile Enforcement Test**

**Vector ID:** B-XP-002

**Inputs:**

- implementation claims Profile III;
- admissibility deterministic;
- enforcement bypass possible.

**Expected Result:**

- Profile III conformance = FAIL because Profile II requirements are inherited.

**B.88 Cross-Profile Replay Test**

**Vector ID:** B-XP-003

**Inputs:**

- implementation claims Profile II;
- replay capability absent.

**Expected Result:**

- Profile II SHALL NOT fail solely because Profile III replay capability is absent.

**B.89 Test Evidence**

For each applicable executed vector, the implementation SHALL preserve evidence sufficient to establish:

- test identifier;
- implementation version;
- configuration;
- governed inputs;
- expected result;
- actual result;
- execution time;
- evaluator or test harness identity;
- PASS or FAIL state;
- discrepancy information where applicable; and
- integrity information where required.

**B.90 Machine-Readable Test Results**

A conformance program MAY require machine-readable test results.

Where provided, machine-readable results SHOULD identify:

- Standard version;
- Annex version;
- test vector identifier;
- Conformance Profile;
- implementation identifier;
- input-set identifier;
- expected-result identifier;
- actual result;
- PASS or FAIL status; and
- result integrity value.

**B.91 Extension Test Vectors**

An implementation or certification program MAY define additional test vectors.

Extension vectors SHALL NOT:

- replace applicable normative vectors;
- weaken expected normative results;
- alter profile inheritance;
- redefine Standard requirements; or
- convert prohibited behavior into conforming behavior.

**B.92 Test Vector Versioning**

Normative test vectors SHALL be version-identifiable.

A material change to:

- input;
- expected result;
- evaluation rule;
- PASS condition; or
- applicable profile

SHALL produce a distinguishable Annex or test-vector version.

**B.93 Test Vector Preservation**

Historical normative test vectors used for a completed conformity assessment SHALL remain reproducibly identifiable.

A later test-vector version SHALL NOT silently replace the test vector used in a prior assessment.

**B.94 Conformance Requirement**

An implementation conforms to this Annex when it:

- executes all applicable normative vectors;
- produces the required governed outcomes;
- demonstrates deterministic repeatability;
- preserves required test evidence;
- satisfies applicable negative tests;
- satisfies inherited lower-profile tests; and
- does not report PASS where required behavior is absent.

**B.95 Canonical Statement**

A conformance claim is credible only when the implementation can be made to fail correctly.

Positive tests demonstrate that permitted action can proceed.

Negative tests demonstrate that non-admissible, non-eligible, stale, revoked, corrupted, or unreproducible action cannot be treated as valid.

Deterministic conformance requires both.

## Annex C — Reference Algorithms and Evaluation Procedures

**Normative**

### C.1 Purpose

This Annex defines normative reference procedures for evaluating Trust-State governed conditions.

The procedures in this Annex establish a common deterministic evaluation model for:

- Rule-State selection;
- Authority-State evaluation;
- evidence sufficiency;
- context and dependency evaluation;
- admissibility;
- Trust-State derivation;
- execution eligibility;
- enforcement;
- continuity;
- revocation;
- recomputation; and
- replay.

A conforming implementation MAY use a different internal algorithm, architecture, or execution model provided that equivalent governed inputs evaluated under equivalent governing conditions produce equivalent governed outputs under the applicable Conformance Profile.

### C.2 Reference Procedure Principle

Reference procedures define required governed behavior.

They do not require a specific:

- programming language;
- software architecture;
- database;
- workflow engine;
- policy engine;
- cryptographic library;
- deployment model; or
- commercial implementation.

Equivalent implementations SHALL preserve the normative decision semantics established by the Standard.

**C.3 Evaluation Inputs**

A governed evaluation SHALL identify the material inputs required for the proposed action.

The evaluation input set SHALL include or reference, where applicable:

- action;
- actor or subject;
- resource or target;
- Authority-State;
- evidence;
- Rule-State;
- context;
- temporal state;
- dependency state;
- revocation state;
- execution conditions; and
- applicable prior Trust-State.

Material inputs capable of altering the governed result SHALL NOT remain undisclosed where independent verification or recomputation is required.

**C.4 Evaluation Boundary**

Before a final determination is issued, the system SHALL establish the governed evaluation boundary.

The boundary SHALL identify, where applicable:

- proposed consequence-bearing action;
- subject;
- target or resource;
- Authority-State scope;
- evidence scope;
- Rule-State scope;
- contextual scope;
- temporal boundary;

- dependency boundary; and
- applicable Conformance Profile.

An evaluation SHALL NOT silently incorporate material conditions outside the governed evaluation boundary.

**C.5 Reference Evaluation Order**

Unless the applicable Rule-State expressly defines another deterministic order, the reference evaluation order SHALL be:

1. identify the proposed action;
2. identify the applicable Rule-State;
3. establish applicable Authority-State;
4. determine authority sufficiency;
5. identify and evaluate required evidence;
6. establish material context and dependencies;
7. evaluate applicable temporal conditions;
8. evaluate applicable revocation state;
9. evaluate applicable constraints;
10. resolve rule precedence, conflict, exceptions, and overrides where applicable;
11. derive admissibility;
12. derive Trust-State;
13. derive execution eligibility where applicable;
14. apply enforcement where applicable; and
15. preserve sufficient evaluation evidence.

A different evaluation order MAY be used where it is deterministic and produces equivalent governed results.

**C.6 Rule-State Selection Procedure**

The applicable Rule-State SHALL be selected deterministically.

Selection SHALL consider, where material:

- action or action class;
- subject or actor class;
- resource;

- jurisdiction;
- policy domain;
- scope;
- effective time;
- version;
- supersession;
- precedence; and
- applicability conditions.

The selected Rule-State SHALL be version-identifiable before a final admissibility determination is issued.

Where no valid Rule-State can be identified, or where materially conflicting Rule-States cannot be deterministically resolved, affirmative admissibility SHALL NOT be established where a valid Rule-State is required.

**C.7 Authority Sufficiency Procedure**

Authority sufficiency SHALL be determined by evaluating whether the applicable Authority-State satisfies the authority conditions required by the Rule-State.

Evaluation SHALL consider, where applicable:

- authority source;
- authorized subject;
- action scope;
- resource scope;
- constraints;
- delegation lineage;
- temporal validity;
- suspension;
- revocation;
- provenance; and
- integrity.

Authority SHALL be SUFFICIENT only where all mandatory authority conditions are satisfied.

Authority SHALL be INSUFFICIENT where a mandatory condition is unsatisfied.

Authority MAY be INDETERMINATE where a required condition cannot be established and the Rule-State permits such classification.

**C.8 Evidence Sufficiency Procedure**

Evidence sufficiency SHALL be determined according to the applicable Rule-State.

Required evidence SHALL be evaluated for, where applicable:

- existence;
- identity;
- validity;
- applicability;
- temporal validity;
- provenance;
- integrity;
- revocation or supersession;
- transformation lineage;
- conflict; and
- contribution to sufficiency.

Evidence SHALL be SUFFICIENT only where every mandatory evidentiary requirement is satisfied.

Evidence SHALL be INSUFFICIENT where a mandatory evidentiary requirement is not satisfied.

Evidence MAY be INDETERMINATE where sufficiency cannot be conclusively established and the Rule-State permits such classification.

**C.9 Governing-Condition Evaluation**

Material governing conditions not fully represented by Authority-State or evidence SHALL be evaluated according to the Rule-State.

Such conditions MAY include:

- context;
- time;
- jurisdiction;
- actor state;
- resource state;
- transaction state;
- dependency state;
- environmental conditions;
- purpose restrictions;

- value limits;
- frequency limits;
- separation-of-duty conditions;
- dual-control requirements;
- exceptions;
- overrides;
- human review;
- probabilistic inputs; and
- other defined constraints.

Unresolved material ambiguity or conflict SHALL NOT silently produce affirmative admissibility.

Where probabilistic or human-derived inputs materially affect the outcome, their permitted role and deterministic interpretation SHALL be defined by the Rule-State.

**C.10 Admissibility Derivation Procedure**

After all mandatory governing conditions have been evaluated, admissibility SHALL be derived.

The reference admissibility procedure is:

**IF**

- required authority is sufficient;
- required evidence is sufficient;
- Rule-State is valid and applicable;
- required context is valid;
- required dependencies are satisfied;
- applicable temporal conditions are satisfied;
- applicable revocation conditions are satisfied;
- applicable constraints are satisfied; and
- no unresolved material conflict or ambiguity remains;

**THEN**

admissibility = **ADMISSIBLE**.

Otherwise, admissibility SHALL be classified according to the applicable Rule-State as:

- NON_ADMISSIBLE;
- INDETERMINATE;

- CONDITIONALLY_ADMISSIBLE;
- DEGRADED; or
- another objectively defined state.

Conditional admissibility SHALL NOT convert failure of a mandatory precondition into permission to execute.

**C.11 Trust-State Derivation Procedure**

Trust-State SHALL be derived from the applicable admissibility determination and governing conditions.

A Trust-State SHALL remain bound to, where applicable:

- action;
- Authority-State;
- evidence set;
- Rule-State;
- context;
- temporal state; and
- determination identity.

Trust-State SHALL NOT be treated as generalized reputation, confidence, or persistent universal trust.

Where materially different governing conditions produce a materially different Trust-State, the resulting Trust-State SHALL be distinguishable from the prior state.

**C.12 Execution Eligibility Procedure**

For Profile II and Profile III, execution eligibility SHALL be derived after admissibility.

The reference eligibility procedure is:

**IF**

- admissibility permits execution;
- Trust-State is valid for the proposed action;
- eligibility conditions remain satisfied;
- no disqualifying material change has occurred;
- no applicable revocation invalidates the determination;

- execution scope matches the governed determination; and
- execution remains within the applicable validity window;

**THEN**

execution eligibility = **ELIGIBLE**.

Otherwise, execution eligibility SHALL be classified according to the applicable Rule-State.

Eligibility SHALL NOT be inferred solely from identity, access, capability, prior authorization, or technical ability to execute.

**C.13 Enforcement and Fail-Closed Procedure**

Where Profile II or Profile III applies, execution eligibility SHALL be verified before the governed action crosses the execution boundary.

The enforcement procedure SHALL verify, where applicable:

1. execution request identity;
2. associated eligibility;
3. action binding;
4. subject binding;
5. resource binding;
6. scope;
7. temporal validity;
8. revocation state;
9. integrity;
10. single-use or reusable status; and
11. material change occurring after eligibility derivation.

Where established eligibility is mandatory, unresolved failure of a mandatory verification condition SHALL result in **DENY**.

Failure to evaluate SHALL NOT be treated as permission to execute.

**C.14 Continuity Evaluation Procedure**

For Profile III, continuity evaluation SHALL determine whether a previously derived Trust-State remains valid following material change.

The reference continuity procedure is:

1. identify prior Trust-State;
2. identify the material change;
3. identify affected governing conditions;
4. identify the applicable Rule-State;
5. determine whether re-evaluation is required;
6. re-evaluate affected conditions;
7. derive updated admissibility;
8. derive updated Trust-State;
9. derive updated execution eligibility where applicable;
10. propagate resulting state where required; and
11. preserve continuity evidence.

A prior valid Trust-State SHALL NOT establish indefinite future validity.

**C.15 Material Change Procedure**

A change SHALL trigger continuity evaluation where it can materially alter:

- authority sufficiency;
- evidence sufficiency;
- Rule-State applicability;
- context validity;
- dependency satisfaction;
- temporal validity;
- admissibility;
- Trust-State;
- execution eligibility; or
- required enforcement behavior.

The applicable Rule-State SHALL define or support determination of materiality.

**C.16 Revocation and Propagation Procedure**

Where revocation affects a governed input or state, the system SHALL:

1. identify the revocation;
2. verify its source and authority where required;
3. identify the affected object;
4. establish scope;
5. establish effective time;
6. identify dependent states;
7. determine propagation requirements;
8. update affected Trust-State;
9. update execution eligibility;
10. enforce the resulting state; and
11. preserve revocation lineage.

Propagation SHALL stop at a dependency boundary only where the Rule-State establishes that the downstream state remains independently sufficient.

**C.17 Restoration Procedure**

A suspended, degraded, expired, or revoked state SHALL NOT automatically return to valid state.

Where restoration is permitted, the system SHALL:

1. identify restoration authority;
2. identify required evidence;
3. verify current governing conditions;
4. apply the current Rule-State;
5. derive the restored or successor state;
6. assign distinguishable identity where material;
7. preserve prior state history; and
8. preserve restoration provenance.

Restoration SHALL NOT erase prior suspension, expiration, degradation, or revocation history.

**C.18 Historical Replay Procedure**

Historical replay SHALL reconstruct a prior determination using the governing conditions applicable at the original evaluation point.

The reference replay procedure is:

1. identify the historical determination;
2. identify the historical action;
3. retrieve or reproduce historical Authority-State;
4. retrieve or reproduce historical evidence;
5. retrieve or reproduce historical Rule-State;
6. retrieve or reproduce historical context;
7. retrieve or reproduce historical dependency state;
8. retrieve or reproduce historical revocation state;
9. identify the historical evaluation procedure;
10. identify the historical canonicalization version;
11. evaluate under historical conditions;
12. derive the replay result; and
13. compare the replay result with the original determination.

A current Rule-State SHALL NOT silently replace the historical Rule-State.

**C.19 Current Re-Evaluation Procedure**

Current re-evaluation SHALL use governing conditions applicable at the current evaluation point.

The procedure SHALL identify, where applicable:

- current action;
- current Authority-State;
- current evidence;
- current Rule-State;
- current context;
- current dependencies;
- current temporal state;
- current revocation state; and
- applicable constraints.

Current re-evaluation SHALL NOT be represented as historical replay.

A current result MAY legitimately differ from a historical replay result where governing conditions differ.

**C.20 Independent Recomputation Procedure**

Where independent recomputation is required, the reference procedure is:

1. obtain preserved governed inputs;
2. verify input identity and integrity;
3. identify the applicable procedure version;
4. identify the applicable canonicalization version;
5. independently canonicalize governed inputs where required;
6. independently evaluate governing conditions;
7. independently derive admissibility;
8. independently derive Trust-State;
9. independently derive execution eligibility where applicable;
10. independently derive applicable integrity values; and
11. compare outputs using defined equivalence criteria.

Recomputation SHALL NOT depend solely upon the original evaluator's claimed result.

**C.21 Recomputation Discrepancy Handling**

A recomputation discrepancy exists where independently derived output does not satisfy the applicable equivalence criteria.

A discrepancy SHOULD be classified according to its material cause, including where applicable:

- input mismatch;
- Authority-State mismatch;
- evidence mismatch;
- Rule-State mismatch;
- context mismatch;
- procedure mismatch;
- canonicalization mismatch;
- integrity mismatch;
- hidden state;
- nondeterministic behavior; or
- implementation defect.

A material discrepancy SHALL NOT be silently ignored.

### C.22 Serialization and Integrity Verification Procedure

Where canonical serialization is required, governed objects SHALL be serialized according to Annex A before generation or verification of applicable integrity values.

The reference procedure is:

1. identify object type;
2. identify canonicalization version;
3. establish canonicalization boundary;
4. normalize values as required;
5. apply deterministic ordering;
6. encode values;
7. serialize;
8. generate or verify the applicable integrity value; and
9. preserve representation identity.

Equivalent governed inputs SHALL produce equivalent canonical representations under the applicable serialization specification.

### C.23 Reference State Machine

A conforming implementation MAY express Trust-State processing through a state machine.

A reference progression MAY include:

**REQUESTED → EVALUATING → ADMISSIBLE → ELIGIBLE → EXECUTED**

with alternate governed states including:

- NON_ADMISSIBLE;
- INDETERMINATE;
- CONDITIONALLY_ADMISSIBLE;
- NOT_ELIGIBLE;
- SUSPENDED;
- EXPIRED;
- REVOKED;
- DEGRADED; and
- TERMINATED.

The state-machine representation is illustrative and SHALL NOT override the normative requirements of the applicable Conformance Profile.

**C.24 Invalid Transition Prevention**

A conforming implementation SHALL prevent invalid governed transitions capable of improperly creating valid Trust-State or execution eligibility.

Invalid transitions include, where applicable:

- NON_ADMISSIBLE → ELIGIBLE without a new governed determination;
- REVOKED → VALID without governed restoration;
- EXPIRED → ELIGIBLE without required re-evaluation;
- INDETERMINATE → EXECUTED where affirmative eligibility is mandatory; and
- SUSPENDED → ACTIVE without governed restoration.

**C.25 Deterministic Error and Unknown-State Handling**

Errors capable of affecting the governed result SHALL have deterministic treatment.

The applicable Rule-State or evaluation procedure SHALL define whether such a condition results in:

- NON_ADMISSIBLE;
- INDETERMINATE;
- SUSPENDED;
- DEGRADED;
- DENY;
- retry;
- escalation; or
- another objectively defined state.

UNKNOWN or INDETERMINATE SHALL NOT be treated as affirmative satisfaction of a mandatory governing condition.

**C.26 No Hidden Decision Logic**

Material decision logic SHALL be represented by the applicable Rule-State, evaluation procedure, or another governed specification.

Where independent recomputation is required, an implementation SHALL NOT depend upon undocumented material state or undisclosed decision logic capable of altering the governed result.

Material external dependencies capable of altering the determination SHALL be identified or reproducibly represented within the applicable evaluation boundary.

**C.27 Cross-Implementation Equivalence**

Independent conforming implementations using equivalent governed inputs, equivalent Rule-State, and equivalent evaluation procedures SHALL produce equivalent governed outputs.

Internal differences in:

- programming language;
- platform;
- architecture;
- storage model;
- deployment environment; or
- implementation technique

SHALL NOT constitute non-conformity where normative output equivalence is preserved.

**C.28 Procedure Versioning and Testability**

Evaluation procedures SHALL be version-identifiable where procedural changes can alter governed output.

Historical procedure versions used for completed determinations SHALL remain reproducibly identifiable within the applicable retention boundary.

Procedures used to establish conformity SHALL be testable against applicable Annex B requirements.

An implementation SHALL NOT be considered procedurally equivalent solely because its architecture appears conceptually similar where required test outcomes cannot be demonstrated.

**C.29 Conformance**

An implementation conforms to this Annex where its governed evaluation procedures:

- identify material governed inputs;
- establish a defined evaluation boundary;
- apply Rule-State deterministically;
- evaluate authority and evidence sufficiency;
- evaluate material governing conditions;
- derive admissibility;
- derive Trust-State;
- derive execution eligibility where applicable;
- enforce execution where applicable;
- govern continuity and revocation where applicable;
- support recomputation and replay where applicable;
- preserve required provenance;
- prevent invalid state transitions;
- provide deterministic failure behavior; and
- produce results equivalent to the normative procedures defined by this Annex.

**C.30 Canonical Statement**

A deterministic standard does not require every implementation to use the same code.

It requires every conforming implementation to preserve the same governed meaning.

Equivalent inputs, evaluated under equivalent governing conditions and rules, SHALL produce equivalent governed results.

## Annex D — Interoperability and Machine-Readable Artifacts

**Normative**

### D.1 Purpose

This Annex defines requirements for machine-readable Trust-State artifacts and interoperable exchange of governed state between conforming implementations.

### D.2 Interoperability Principle

A conforming implementation MAY use any internal architecture provided that externally exchanged governed artifacts preserve the identities, versions, boundaries, and integrity information required to interpret them deterministically.

#### D.3 Artifact Classes

Machine-readable artifacts MAY represent:

- Authority-State;
- evidence or evidence-set identity;
- Rule-State;
- admissibility determination;
- Trust-State;
- execution eligibility;
- continuity or revocation state;
- conformance claim; and
- certification record.

#### D.4 Required Artifact Identity

Each exchanged governed artifact SHALL identify or reference, where applicable:

- artifact type;
- artifact identifier;
- applicable specification version;
- governing object version;
- creation or effective time;
- applicable scope or boundary; and
- integrity information.

**D.5 Reference Binding**

An artifact that depends upon another governed object SHALL identify that dependency sufficiently to prevent material ambiguity.

References SHOULD include the referenced object identifier and integrity value where integrity verification is required.

**D.6 Admissibility Artifact**

A machine-readable admissibility artifact SHALL identify or reference:

- action;
- Authority-State;
- evidence set;
- Rule-State;
- material context;
- admissibility outcome;
- Trust-State where applicable;
- evaluation-procedure version; and
- integrity information.

**D.7 Execution Eligibility Artifact**

An execution eligibility artifact SHALL identify or reference:

- admissibility determination;
- eligible action;
- actor or subject;
- resource or scope;
- eligibility state;
- validity period;
- applicable conditions;
- revocation state; and
- integrity information.

**D.8 Continuity and Revocation Artifact**

A continuity or revocation artifact SHALL identify or reference:

- affected state;

- material change or revocation;
- effective time;
- scope;
- resulting state;
- applicable Rule-State; and
- propagation effects where applicable.

**D.9 Conformance Claim Artifact**

A machine-readable conformance claim SHALL identify:

- Trust-State Standard version;
- Conformance Profile;
- implementation identifier;
- conformance boundary identifier;
- assessment status;
- assessment date;
- applicable specification versions; and
- supporting evidence or certification reference where applicable.

**D.10 Certification Record Artifact**

A machine-readable certification record SHALL identify:

- certification identifier;
- certified implementation;
- Standard version;
- Conformance Profile;
- certified boundary;
- certification body;
- certification status;
- issue date;
- expiration or renewal condition where applicable; and
- integrity information.

**D.11 Machine-Representable Conformance Boundary**

Where a machine-readable conformance claim is used, the associated conformance boundary SHOULD be machine-representable.

For Profile III certification, the applicable certification program MAY require machine-representable boundary information sufficient to identify:

- action classes;
- execution boundary;
- Rule-State boundary;
- continuity boundary;
- replay boundary; and
- material dependencies.

**D.12 Interoperability**

Two conforming systems exchanging a governed artifact SHALL be able to determine:

- what object is represented;
- which specification version applies;
- what boundary applies;
- which referenced objects were relied upon;
- whether integrity verifies; and
- whether the artifact is applicable to the receiving system's governed determination.

**D.13 Semantic Preservation**

Transformation between supported artifact formats SHALL preserve governed meaning.

A transformation SHALL NOT:

- alter material identifiers;
- alter Rule-State version;
- alter authority scope;
- omit material evidence references;
- alter determination outcome;
- expand eligibility; or
- alter integrity-protected content without producing a distinguishable successor artifact.

**D.14 Unknown or Unsupported Artifact**

An artifact that cannot be interpreted under an applicable supported specification SHALL NOT be presumed valid.

**D.15 Conformance**

An implementation conforms to this Annex where machine-readable governed artifacts within its declared interoperability boundary are:

- version-identifiable;
- scope-identifiable;
- deterministically interpretable;
- integrity-verifiable where required;
- capable of preserving material references; and
- interoperable without silent alteration of governed meaning.

**D.16 Canonical Statement**

Machine-readable interoperability does not require identical systems.

It requires that governed meaning survive exchange.

**Annex E — Certification and Assessment Procedures**

**Normative for Trust-State certification programs**

**E.1 Purpose**

This Annex defines minimum procedural requirements for conformity assessment and certification under the Trust-State Standard.

This Annex governs certification administration and does not create additional implementation requirements beyond the applicable Conformance Profile.

**E.2 Assessment Principle**

Certification SHALL be based upon objectively reviewable evidence demonstrating conformity to:

- an identified Trust-State Standard release;
- an identified Conformance Profile; and
- a declared conformance boundary.

**E.3 Assessment Sequence**

A certification assessment SHALL include:

1. scope and boundary confirmation;
2. implementation identification;
3. applicable Profile identification;
4. normative requirement mapping;
5. evidence review;
6. applicable Annex B testing;
7. non-conformity identification;
8. remediation review where applicable;
9. certification decision; and
10. issuance of a certification record.

**E.4 Assessment Evidence**

The assessor SHALL obtain sufficient evidence to determine whether applicable normative requirements are satisfied.

Evidence SHALL be traceable to the assessed implementation and configuration.

### E.5 Test Requirements

Applicable Annex B vectors SHALL be executed or otherwise demonstrated through an approved equivalent test procedure.

A certification program MAY define additional tests but SHALL NOT weaken normative expected outcomes.

### E.6 Non-Conformity

Failure of a mandatory requirement SHALL be recorded as a non-conformity.

The certification program SHALL define criteria for:

- remediation;
- retesting;
- conditional status where permitted;
- suspension; and
- denial of certification.

### E.7 Certification Decision

The certification decision SHALL identify:

- implementation;
- Standard version;
- Conformance Profile;
- certified boundary;
- assessment outcome;
- material limitations;
- certification status; and
- certification identifier.

### E.8 Assessor and Decision Independence

Where required by the certification program, the certification decision SHALL be performed by a person or function sufficiently independent from implementation activity and evidence production.

**E.9 Certification Record**

The certification record SHALL contain or reference the information required by Section 15.

The record SHOULD be independently verifiable.

**E.10 Surveillance and Change**

Material change to a certified implementation SHALL be evaluated according to Sections 15 and 17.

The certification program SHALL define when change requires:

- notification;
- targeted assessment;
- full reassessment;
- suspension; or
- renewal.

**E.11 Renewal**

Certification renewal SHALL confirm continued conformity within the certified boundary.

Renewal SHALL consider material changes, prior non-conformities, current test results, and certification status.

**E.12 Suspension, Withdrawal, and Revocation**

Certification status changes SHALL be governed and recorded.

A suspended, withdrawn, expired, or revoked certification SHALL NOT be represented as active.

**E.13 Appeals**

The certification program SHOULD provide a documented appeal mechanism for certification decisions.

Appeal handling SHALL preserve reviewer independence.

**E.14 Certification Mark**

Where a certification mark is authorized, use SHALL be limited to the certified implementation, Profile, boundary, and active certification period.

**E.15 Confidential Evidence**

Certification evidence MAY remain confidential where public disclosure is not required.

Confidentiality SHALL NOT eliminate the requirement for sufficient assessor access and verification.

**E.16 Assessment Record Retention**

Assessment and certification records SHALL be retained for the period defined by the certification program.

**E.17 Conformance**

A certification program conforms to this Annex where certification decisions are:

- evidence-based;
- profile-specific;
- boundary-specific;
- test-supported;
- independently reviewable;
- lifecycle-governed; and
- protected against misleading use.

**E.18 Canonical Statement**

Certification is a governed determination of demonstrated conformity.

It does not create the capability it certifies.

**Annex F — Consequence Science and External Framework Mapping**

**Informative**

**F.1 Purpose**

This Annex provides informative context regarding the conceptual foundation of the Trust-State Standard and its potential relationship to external governance, assurance, regulatory, and technical frameworks.

This Annex is informative.

It does not create implementation or conformance requirements.

**F.2 Consequence Science**

The Trust-State Standard is informed by the premise that consequence is the foundational condition giving rise to governance.

Consequence is not itself an authorization state, trust state, rule, or execution decision.

Governance operates above that foundational condition by determining whether a proposed consequence-bearing action satisfies the conditions required to proceed.

**F.3 Layer 0**

Within the associated Consequence Science model, Layer 0 represents **Consequence**.

Layer 0 expresses the condition that an action, pathway, or state transition may produce consequence.

The Trust-State Standard does not require implementations to predict or guarantee consequence.

Instead, it governs the admissibility and execution conditions applied before consequence-bearing action proceeds.

**F.4 Relationship to Trust-State**

The conceptual sequence may be represented informatively as:

**Consequence → Governing Conditions → Admissibility → Trust-State → Execution Eligibility → Execution**

Continuity and revocation govern whether previously established states remain sufficient as material conditions change.

Independent recomputation and replay support later verification of the governed determination.

**F.5 Consequence and Outcome**

TSS conformity does not establish that a resulting consequence is:

- desirable;
- ethical;
- lawful;
- optimal;
- safe; or
- socially beneficial.

Those determinations may be governed by external policy, law, regulation, or domain-specific requirements.

TSS governs whether declared governing conditions were deterministically applied.

**F.6 External Framework Mapping**

Trust-State requirements MAY be mapped to external frameworks for purposes such as:

- regulatory analysis;
- assurance;
- procurement;
- cybersecurity governance;
- AI governance;
- audit;
- risk management; and
- conformity assessment.

Such mappings are informative unless expressly incorporated into a separate normative program.

**F.7 Regulatory Mapping**

A regulatory mapping MAY identify how TSS capabilities support external requirements involving:

- authorization;

- evidence;
- traceability;
- human oversight;
- execution controls;
- change management;
- logging;
- accountability;
- verification; and
- lifecycle governance.

A mapping SHALL NOT be interpreted as a declaration of regulatory compliance unless such compliance has been separately established.

**F.8 Standards Mapping**

TSS MAY be mapped to technical or governance standards maintained by other standards organizations.

Mapping SHOULD distinguish:

- equivalent concepts;
- complementary concepts;
- partial overlap;
- unsupported requirements; and
- requirements unique to TSS.

**F.9 Jurisdiction Neutrality**

External mappings do not alter the jurisdiction-neutral character of the Trust-State Standard.

Jurisdiction-specific requirements may be represented as Rule-State, Authority-State, evidence, context, or other governed inputs without becoming universal TSS requirements.

**F.10 Informative Diagrams**

Informative diagrams MAY be published to illustrate:

- Layer 0;
- the consequence-to-execution sequence;
- Conformance Profiles;
- Authority-State relationships;
- replay and recomputation;

- continuity and revocation; and
- external framework mappings.

Diagrams SHALL NOT override normative text.

**F.11 Relationship to Future Guidance**

Implementation guides, regulatory mappings, sector profiles, and educational material MAY build upon this Annex.

Such material remains informative unless separately designated as normative through the applicable governance process.

**F.12 Canonical Statement**

Consequence is the reason governance exists.

The Trust-State Standard does not govern which consequences society should choose.

It governs whether the conditions permitting consequence-bearing digital action can be established, enforced, and independently verified.